2013-09

Content-aware adaptive compression of satellite imagery using artificial vision

Wilcox, Jeffrey

Monterey, California: Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CONTENT-AWARE ADAPTIVE COMPRESSION OF SATELLITE IMAGERY USING ARTIFICIAL VISION

by

Jeffrey Wilcox

September 2013

Thesis Advisor: Mathias Kölsch
Second Reader:: Richard Olsen

This thesis was performed at the MOVES Institute
Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
09-27-2013	Master's Thesis	06-24-2011—09-27-2013

4. TITLE AND SUBTITLE

CONTENT-AWARE ADAPTIVE COMPRESSION OF SATELLITE IMAGERY USING ARTIFICIAL VISION

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Jeffrey P. Wilcox

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of the Navy

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.IRB Protocol Number:N/A

14. ABSTRACT

This thesis aims to improve image throughput from satellite to Earth by using Artificial Vision to perform data compression before the downlink.Onboard Analysis for Selective Imagery Compression (OASIC) is a hybrid compression algorithm designed for oceanic imagery, incorporating both lossless and lossy compression methods to achieve a high compression ratio with minimal noise on vessels of interest.This is achieved by separating the vessels from the surrounding ocean and storing them with high fidelity, while compressing the remainder of the image with low fidelity.The performance of OASIC is examined on full resolution panchromatic satellite images and compared to both lossless and lossy JPEG2000 compressed images.In nearly all configurations tested, OASIC outperforms JPEG2000, achieving an average fifteen-fold improvement in compression ratios while maintaining a nearly lossless fidelity for the vessels within the OASIC compressed images.This content-sensitive compression algorithm can potentially enable the transmission of higher spatial resolution images, with more spectral bands, and at higher download speeds from space.

15. SUBJECT TERMS

Lossless Image Compression, Lossy Image Compression, Ship Detection, Support Vector Machine, Discrete Wavelet Transform, Artificial Vision, Satellite Imagery

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
Unclassified	Unclassified	Unclassified	UU	77	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 8–98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

CONTENT-AWARE ADAPTIVE COMPRESSION OF SATELLITE IMAGERY USING ARTIFICIAL VISION

Jeffrey P. Wilcox
Lieutenant, United States Navy
B.S., United States Naval Academy, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SPACE SYSTEMS OPERATIONS

from the

NAVAL POSTGRADUATE SCHOOL
September 2013

Author: Jeffrey P. Wilcox

Approved by: Mathias Kölsch
Thesis Advisor

Richard Olsen
Second Reader

Rudy Panholzer
Chair, Department of Space Systems

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis aims to improve image throughput from satellite to Earth by using Artificial Vision to perform data compression before the downlink. Onboard Analysis for Selective Imagery Compression (OASIC) is a hybrid compression algorithm designed for oceanic imagery, incorporating both lossless and lossy compression methods to achieve a high compression ratio with minimal noise on vessels of interest. This is achieved by separating the vessels from the surrounding ocean and storing them with high fidelity, while compressing the remainder of the image with low fidelity. The performance of OASIC is examined on full resolution panchromatic satellite images and compared to both lossless and lossy JPEG2000 compressed images. In nearly all configurations tested, OASIC outperforms JPEG2000, achieving an average fifteen-fold improvement in compression ratios while maintaining a nearly lossless fidelity for the vessels within the OASIC compressed images. This content-sensitive compression algorithm can potentially enable the transmission of higher spatial resolution images, with more spectral bands, and at higher download speeds from space.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

C/R Compression Ratio
dB Decibels
DCT Discrete Cosine Transform
DNC Digital Nautical Chart
DWT Discrete Wavelet Transform
DSP Digital Signal Processor
FPGA Field-Programmable Gate Array
IEC International Electrotechnical Commission Organization for Standardization
IEEE Institute of Electrical and Electronics Engineers
ISO International Organization for Standardization
ITU International Telecommunication Union
JP2 Minimal JPEG 2000 [filetype]
JPEG Joint Pictures Expert Group [filetype]
LEO Low Earth Orbit
MSE Mean Square Error
OAI OASIC Aggregated Image [filetype]
OASIC Onboard Analysis for Selective Imagery Compression
PNG Portable Network Graphic [filetype]
PSNR Peak Signal to Noise Ratio
RMSE Root Mean Square Error
ROC Receiver Operating Characteristic
SPAWAR Space and Naval Warfare Systems Command
SVM Support Vector Machine
VQ Vector Quantization

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgements

I would like to express my gratitude to my thesis advisor, Associate Professor Mathias Kölsch, for his invaluable assistance in the writing of this thesis, and for introducing me to the vast world of Artificial Vision. Furthermore, I would like to thank Professor Richard Olsen for introducing me to the world of Remote Sensing and providing me the motivation to improve digital imagery compression. Also, I would like to express my gratitude to the Space Systems Academic Group and Doctor Rudy Panholzer, as well as the United States Navy for constantly motivating me to dare greatly, go forth and achieve. Finally, I want to thank Katie Rainey of Space and Naval Warfare Systems (SPAWAR) for providing me the pristine satellite images from which all of my experiments, results and conclusions are derived.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

In this chapter, the motivation for the development of what is called the OASIC algorithm (pronounced "oasis") and the problem it aims to solve are discussed. The goals of this research are to apply artificial vision to digital imagery compression and to compare its performance to conventional image compression methods.

1.1 Background

When speaking over a radio, it is considered good practice to keep your report brief, to the point and avoiding any unnecessary transmission so that one does not inadvertently tie up the scarce resources of the radio network. Conservation of channel capacity, as a critical resource, is mandatory for satellite communications to the Earth due to both the limited transmit power of the satellite as well as the increasingly crowded spectrum used by satellites in space. An additional hurdle is the satellite may be operating in a contested environment where capacity is severely reduced.

The Onboard Analysis for Selective Imagery Compression algorithm (OASIC) aims to conserve satellite channel capacity when transmitting oceanic imagery to Earth. OASIC conserves channel capacity by improving data compression by assuming the only objects within the image that require high fidelity are ships. Through the use of artificial vision, OASIC attempts to classify all pixels within an image as either ship or other, less important characteristics such as waves, visible seabed, clouds and other such phenomena.

1.2 Satellite Imagery

The concept of acquiring imagery from above dates back to antiquity where scouts would climb the high peaks overlooking a rival camp to gather intelligence or climb a tree to help navigate through rough terrain.

As technology improved, so too did the altitude of the observer. From hot air balloons to hydrogen filled dirigibles to high altitude aircraft such as the U-2 and finally to orbiting satellites the quest to see more has driven the observer from the atmosphere and into orbit. Satellite-borne observation has its roots in the late 1950s era Corona program developed by the United States, which used analog film cameras and airdropped canisters to return imagery to Earth.

The transition away from film to radio signals and eventually to digital transmissions cemented the imaging satellite's presence in outer space.

Advancements in optical detectors allow for ever higher image resolutions and the detected spectra can now span from far infrared to ultraviolet, with multiple polarizations. With more resolution and spectra, however, more channel capacity is required to send the information to the Earth.

1.3 Downlink Limitations

Transmissions from a satellite to the surface of the Earth are referred to as *downlink*.

The first limitation the downlink faces is power. Imaging satellites are typically solar powered, and require ever larger and more elaborate solar arrays to generate sufficient power to keep up with the demands of their various powered systems including the transmitter. Highly successful commercial imaging satellites such as World View-2 require a large 3.2kW solar array to provide enough power to operate.

The second limitation is signal noise. Earth, the location of the receiver, is an electromagnetically noisy environment and the satellite itself must contend with its own internal electronic noise as well as signal distortions induced by natural radiation in space.

Satellites are also restricted by mass and physical dimensions, limiting the transmitting antenna dish area and necessitating ever more creative methods of collapsible antennas to push the envelope. World View-2 weighs 3.2 tons, with much of that mass dedicated to power.

The transmission carrier to noise ratio (C/N_0) is defined in Equation 1.1 and computed with the gain of the transmitter dish A_t, its power P_t, and gain of the receiver dish (A_r). K is Boltzmann's constant, the temperature T_e of the transmitter (in Kelvin), and L_p and L_d are free space and atmospheric losses, respectively.

$$\frac{C}{N_0} = \frac{A_t P_t (L_p L_d) A_r}{K T_e} \tag{1.1}$$

Channel capacity, is the rate at which bits can be propagated through the range of frequencies used by the satellite and is calculated by Shannon's Limit shown in Equation 1.2. Channel capacity I is measured in bits per second (bps) and is directly proportional both frequency bandwidth B in Hertz and the carrier to noise ratio calculated above.

$$I < B \cdot Log_2 \cdot (1 + \frac{C}{N}) \tag{1.2}$$

A typical imagery satellite such as Digital Globe's World View-2 satellite orbits at an altitude of 770 km, in what is known as Low Earth Orbit (LEO). LEO offers the closest view of the Earth, improving image resolution but limiting the time the satellite is able to downlink its images to any particular ground station. The orbital period for LEO (the time it takes to complete a single orbit) is measured in minutes (100 minutes for World View-2, for instance), with a receiving station only in view for a small fraction of that time. Time is the final limitation, and can be mitigated by the addition of more ground stations, increased channel capacity or relaying the transmission through other satellites.

A satellite such as World View-2 captures up to 331 Gbits of imagery on a single orbit, but requires an 800 Mbps downlink throughput to transmit the data to Earth. Any data not able to downlink may have to be stored in a finite on-board storage and wait, up to an hour, to resume the downlink. These limitations only grow more pronounced as technology continues to improve and satellites demand more channel capacity than the solar arrays, antennas and low-noise amplifiers can provide.

One promising solution is to improve data compression and use the existing channel capacity more efficiently.

1.4 Data Compression

The concept of data compression revolves around the concept of representing a data set with less bits than the original data represents. Lossless data compression reduces the amount of bits needed to represent data by taking advantage of statistical redundancy within the source data. The original data is reconstituted entirely with no errors when lossless compression is used.

Lossy data compression, however, takes advantage of the relative importance of some data over other and aims to quantize or remove the less important data. For the popular lossy music compression standard MPEG Layer 3 (MP3) the audio frequencies and tonal components of

the audio outside normal human perception are removed or reduced, leading to tremendous compression efficiency with little to no perceived loss of quality.

JPEG, the de-facto graphical image standard used by the World Wide Web, similarly takes advantage of the limits of human perception by reducing the fidelity of the color space while preserving the luminosity.

OASIC is a lossy image compression algorithm that aims to preserve the quality of the vessels while sacrificing everything else. It is also intended to ultimately be implemented aboard imaging satellites, and be able to operate within the memory and processor constraints dictated by their architecture.

1.5 Research Goals

The purpose of this research is to validate the concept of Content-Aware Adaptive Compression of Satellite Imagery Using Artificial Vision. The OASIC algorithm is used to compress and uncompress actual satellite images in order to analyze the compression performance and fidelity losses. This research aims to show that OASIC not only compresses oceanic satellite images better than contemporary compression techniques such as JPEG2000, but also does so with less degradation to the vessels within the images.

CHAPTER 2:
Related Work

In this chapter, the methods of feature extraction, classification, and compression are discussed. The goal of OASIC is to reduce the amount of channel capacity consumed by an orbiting imaging satellite when transmitting captured images to the surface. Artificial vision based ship detection algorithms are well researched, and there are several examples that share similarities to the OASIC algorithm. The area of digital data compression, especially image compression, is also well researched. The OASIC algorithm incorporates these two distinct topics.

2.1 Low Level Feature Extraction

Low level features are the smallest units of information of an image that are read directly from the digital medium.

2.1.1 Discrete Wavelet Transform

In the area of Computer Vision, there are many proven methods of low level feature extraction. OASIC uses the Discrete Wavelet Transform (DWT), as according to Meyer [1], it takes advantage of the relatively low energy of ocean texture compared to ship texture in the frequency domain yielded by the wavelet transform. The DWT is also adept at extracting desired objects from images saturated with noise as described by Casasent [2].

The wavelet decomposition as described by Antonini [3] acts as a two-dimensional digital high-pass filter, removing all of the subtle changes in pixel intensity associated with ocean wave tops. This leaves only the features that abruptly differ from their neighboring environment. In effect, wavelet decomposition suppresses much of the natural ocean, while expressing the objects on the surface.

As described by Tello [4] and Selvi [5], three of the four sub-band products of the DWT (HH, HL and LH) can be used to localize, down to a pixel, the existence of an object within a noisy background. According to both Tello [4] and Strickland [6], the Discrete Wavelet Transform is well suited for detecting edges in a noisy image because it natively suppresses noise. However, edges may not stand out against the noise at all resolutions, therefore multiple recursive wavelet decompositions may be required to detect a wide range of object sizes, forming a pyramid as described by Bogush [7].

Huang [8] suggests that the optimal number of wavelet pyramid octaves is three to four. Experiments with OASIC detection and classification have experimentally determined that three octaves is the optimal number, agreeing with Huang's research. Kiely [9] and Zhu [10] use this number of pyramid octaves as well.

Tello's use of wavelet decomposition differs from OASIC's feature extraction in that it applies the correlation of a 4th sub-band (LL). In the case of OASIC, this sub-band is still decomposed to form the next octave of the pyramid, but is not directly supplied to the classifier for analysis.

In the case of Tello, Corbane, Fang [11] and Huang [8], their papers include some type of de-noising stage pre or post feature extraction. This step is absent in OASIC as it expects relatively low energy noise common in optical imagery over much noisier SAR images cited in their work. OASIC also benefits from the inherent de-noising qualities of the DWT.

Experimental findings agree with the research of Tello et. al. in that the DWT handles ocean waves very well. Because OASIC is designed for optical and not SAR imagery, clouds are a concern while radar associated clutter is not. OASIC makes no attempt at masking clouds, however, and relies on the versatility of the DWT to spot vessels through partial cloud cover and ignore large clouds with gradual changes in pixel intensity.

2.2 Feature Classification

Once low level feature extraction has been performed, OASIC combines the outputs of the DWT into an input vector which is fed to a Support Vector Machine for training and classification.

2.2.1 Support Vector Machine

Other ship detection methods have also combined feature extraction methods and learning algorithms for similar detection and compression purposes to OASIC such as Fang [11]. Their research differs in that their learning algorithm is a neural network, and compression is performed by vector quantization. Thus, they do not explore the DWT, SVM and compression algorithm combination that OASIC employs.

The work of Zhu [10] is similar to OASIC in that they use the DWT for feature extraction with the optimal three-octave pyramid and also use an SVM for classification. OASIC differs significantly, however, in that it performs no additional filtering of the DWT products before the classification stage, and accepts a certain number of false positives as inevitable. OASIC also makes no attempt to identify what kind of vessel the object is, its course, or its speed.

Work by Máttyus [12] bears similarities to OASIC as they, too, use the DWT for low level feature extraction and a learning algorithm to perform the classification step. Rather than using water-masking to eliminate surface features from consideration, OASIC uses terrain-masking which is functionally the same method. However, their use of the DWT outputs differ in that the coefficient sub-bands are directly used by a classifier. Instead, their learning algorithm relies on derived Haar-like features and AdaBoost to form their classifier. Their detector performs multiple passes at different rotations, where this step is not needed for OASIC.

Rainey [13] uses a similar combination of feature extraction via wavelets and multiple types of strong and weak classifiers including SVM. OASIC differs in that it solely relies on the DWT for feature extraction and SVM for classification with the goal of facilitating better compression performance. Although not used for ship detection, the methodology of Schneiderman [14] is similar in that the DWT is used for feature extraction and the resultant coefficients are fed into an SVM, albeit with additional processing.

The work of Corbane [15] [16] [17] describes the use of DWT for feature extraction, and also discusses the merits of separating large images into more managable chunks of equal size called tiles. OASIC also uses tiles in the same way, performing the DWT to extract low level features from a single tile, then performing the classification on those features via SVM.

As stated by Degirmenci [18], SVMs can be relied upon to provide excellent classification but care must be taken to select good features for training and classification as SVMs tend to be processing intensive otherwise.

The efforts of Corbane, Máttyus [12], Zhu [10], and Rainey [13] describe a similar method in their works, but differ from OASIC in that their goal is ship detection. OASIC uses ship detection only for the purposes of compression. The general shape of the area encompassing the detected object is not important, and the number of false positives is not as critical to OASIC for this reason.

2.3 Compression

The overarching purpose of OASIC is to reduce the amount of channel capacity needed to downlink a satellite image while retaining high fidelity for ships within an image. To accomplish this, OASIC uses artificial vision to separate vessels within an image and all else remaining into two layers. The first layer, the foreground, contains the detected ships. The second layer, the background layer, contains everything else including the ocean, clouds and any terrain that has not

been removed via terrain mask. Both foreground and background layers are then compressed through conventional means at different fidelity settings.

Work for this purpose is similar to Marcia [19] in that it allows for a high resolution image with pockets of high fidelity to be reconstructed from a sparse dataset. The implementation, however, differs from OASIC in that they do not use detection or artificial vision to define areas of an image that are to compressed with a higher fidelity as OASIC does.

2.3.1 Lossless and Lossy Image Compression

A lossless image is one that contains the exact same pixels before and after being uncompressed. When compressing natural images, there is often a chaotic element that is difficult to compress losslessly and still achieve a reduction in size. This type of compression is invaluable in applications where the pixel values themselves are used to glean additional intelligence from an image, such as reading aircraft markings from a wing of a jet on an aircraft carrier. Such fine details may be obliterated by lossy compression.

In the early 1990s, driven by the emergence of the Internet and the demand for multimedia over a bandwidth-limited dial-up connection, lossy compression became popular in the form of the JPEG standard. Lossy images are compressed image that sacrifice fidelity, often in subtle or imperceptible ways, to create a smaller file than can be achieved with lossless compression alone. Very high compression ratios can therefore be achieved at the cost of fidelity. OASIC uses both of these types of compression: lossless on the foreground, and lossy on the background.

2.3.2 JPEG2000

JPEG2000 is a relatively new compression standard that can compress images in both lossly and lossless modes. This algorithm offers excellent compression performance with a variable level of quality for its lossy mode making it ideal for use in OASIC. The JPEG2000 compression standard as defined by Skodras [20], is used to compress both foreground and background image layers.

He et al. [21] describe the process by which the JPEG2000 algorithm continues to divide an area of an image using a quadtree via successive wavelet decompositions during lossy compression. To minimize the file size of a lossy JPEG2000 image used by OASIC's background, OASIC suppresses detected objects within the image so that it contains only low-frequency ocean pixels. This step prevents the need for additional wavelet decomposition thereby reducing the file size and improving its compression efficiency.

2.3.3 Selective Compression

The method of selective high-fidelity compression described by Mekisso [22] differs in that the coordinates of a bounding box that define the area of high fidelity are provided to the encoding function. OASIC aims to determine the number, size and position of bounding boxes itself using artificial vision. Furthermore, the selective compression performed by OASIC is performed on two images that have been segmented from the same source with different fidelity settings and two different compression methods.

Compression of a composite image of two different layers has been performed by Kiely [9] who used a lossless (JPEG-LS) compression paired with a lossy (JPEG-2000) to obtain similar results, validating the method.

OASIC makes use of efficient packing of rectangles, implementing a derivative of Korf [23] to pack detected objects in preparation for lossless compression. OASIC assumes an optimal rectangle's horizontal width is a multiple 16 to facilitate the most efficient compression.

Compressing images tile-wise is discussed by Fowler [24]. While OASIC does not compress in this manner, it performs the DWT based feature extraction and classification tile-wise at an optimal tile size of 512×512. This method is discussed in greater detail in Chapter 3.

Similar to work demonstrated by Xing [25], OASIC can also compress irregularly shaped objects, though this is done by simply enclosing the irregular shape in a rectangle and setting all non-object pixels to designated transparent pixel value (defaulted to black), or using an alpha channel if all 256 possible pixels are already present in the shape to be compressed.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Methodology

In this chapter, the methods used to implement the preparation, feature extraction, training, classification, and compression are discussed. The basic operation of the OASIC algorithm can be broken up into two major parts as shown in Figure 3.1. The Detection Stage is further broken up into Feature Extraction and Classification.

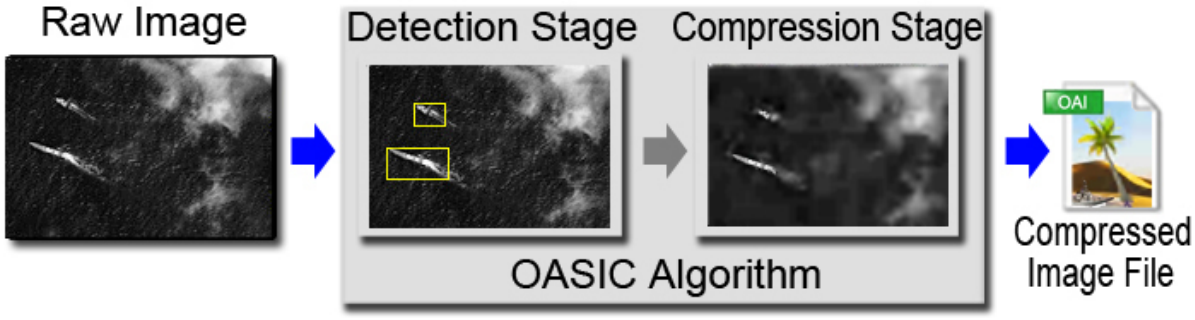

Figure 3.1: Simple representation of the two major stages of OASIC: Detection and Compression.

3.1 Image Preparation

OASIC expects 8-bit per channel panchromatic (grayscale) images. It does, however, support color images, though they are converted to panchromatic and downsampled automatically before testing. The 8-bit, panchromatic image limitation is imposed in order to determine the OASIC's performance when compressing one of the more limited forms of commonly used satellite imagery.

3.1.1 Terrain Masking

The goal of OASIC is to preserve the vessels at sea within an image. It is therefore advantageous to remove any terrain from an image to both prevent it from consuming precious channel capacity, and to prevent the classifier from erroneously detecting ships ashore.

To address this issue, all land terrain is replaced with black, and the bordering ocean texture is faded into the newly erased areas. In this way, the DWT does not produce lines of high energy coefficients at the interface between the blacked out shores and the ocean which may be mistaken for lines of vessels by the classifier.

For the following experiments in Chapter 4, this step is applied manually, however, assuming the satellite's position and camera angle is precisely known, an existing vector based nautical chart can be converted into a mask and used to remove the terrain in order to automate this step.

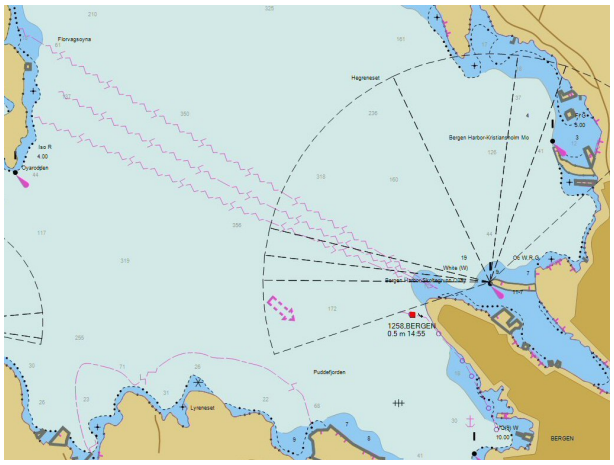

Figure 3.2: An example of a vector-based Digital Nautical Chart (DNC) which could be used to automatically remove most of the terrain from a satellite image.

3.1.2 Converting to Panchromatic

Images may be color but must be converted to an 8-bit channel panchromatic image. Early experimentation indicates there is no difference in performance when using color images that are converted to panchromatic compared to images that are natively panchromatic. It is suspected, though untested, that IR or hyper-spectral images would work as well.

Figure 3.3: The original unprepared image (left) is converted to single channel panchromatic and the terrain is removed (right).

3.1.3 Bit Depth Scaling

The OASIC algorithm is designed to compress 8-bit images only. Any images with a larger bit depth are scaled to 8-bit before they are processed.

3.2 Feature Extraction

Feature extraction is a necessary step in preparing the image for training and prediction by a classifier. Simply loading the raw pixel data into a classifier algorithm is seldom effective or efficient. Instead, feature extraction aims to obtain information about not only the pixel, but the pixel's interactions with its surroundings that differentiate the objects within the image. The classifier uses these differentiating features to attempt to separate the objects from their background. The features may constitute a smaller set of data than the raw pixel data, but this is not always the case: OASIC's feature dataset is often many times larger.

3.2.1 Tiling

A tile is a smaller subset of the larger image. OASIC examines the given image one tile at a time beginning at the upper left corner of the image and ending at the lower right corner. Each tile is square, comprising 512×512 pixels. If the image dimensions are not multiples of 512, OASIC automatically pads the image accordingly with copies of adjacent pixels.

3.2.2 Discrete Wavelet Transform

OASIC uses the DWT to extract the necessary features from each 512×512 tile. Each wavelet decomposition produces four coefficient matrices called sub-bands. The DWT was chosen for feature extraction for its native ability to separate low frequency waves from high frequency waves such as the edges separating ocean from ship. The DWT is defined in Equation (3.1) where W_f is the resultant coefficient matrix of the input image f and mother wavelet function $\phi(x,y)$. The parameters are s for scale, and $t = (t_x, t_y)$. Equation (3.2) defines the mother wavelet function. The algorithm used for applying the DWT to a two dimensional images or matrix is described by Mallat [26].

$$W_f(s,t_x,t_y) = [f(x,y) \cdot \phi_{s,t}(x,y)] \tag{3.1}$$

$$\phi_{s,t}(x,y) = \frac{1}{\sqrt{s}} \cdot \phi\left(\frac{x-t_x}{s}, \frac{y-t_y}{s}\right) \tag{3.2}$$

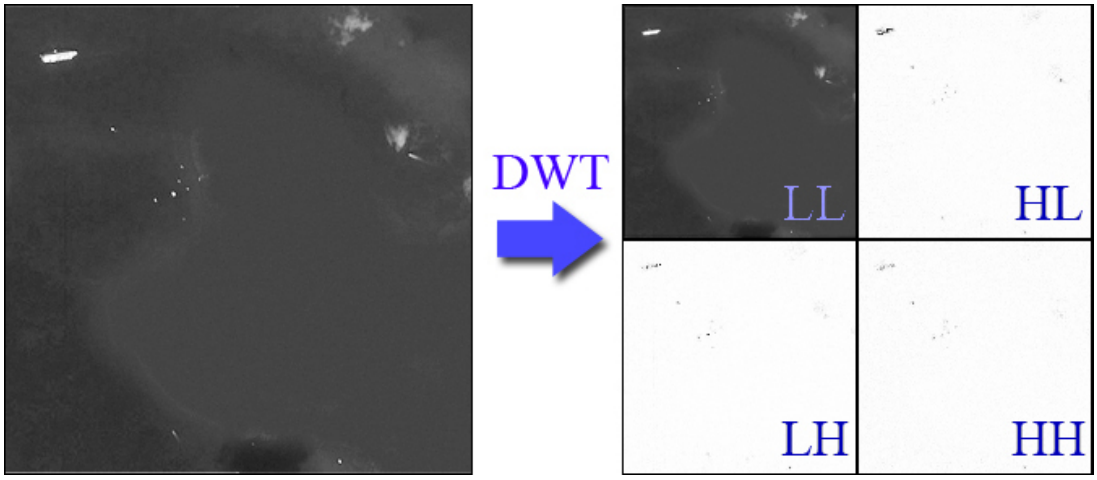

Figure 3.4: An image is decomposed via DWT to produce four 4 sub-bands. Note: LL is a half-scale version of the original.

3.2.3 Sub-bands

Each of the four sub-bands are unique: LL is the low-pass matrix of the source tile, HL is the horizontal coefficient matrix, LH is vertical coefficient matrix and HH is the diagonal (upper-left to lower-right) coefficient matrix. Each sub-band is half the dimensions of the source tile along both x and y axes. Therefore, after a single decomposition, all sub-band matrices contain 256×256 coefficients.

$$f^{LL(g-1)}(i,j) = \sum_{k_1=0}^{L-1} \left(\sum_{k_2=0}^{L-1} f^{LL(g)}(2i+k_1, 2j+k_2) \cdot l_{k_2} \right) \cdot l_{k_1} \tag{3.3}$$

$$f^{HL(g-1)}(i,j) = \sum_{k_1=0}^{L-1} \left(\sum_{k_2=0}^{L-1} f^{LL(g)}(2i+k_1, 2j+k_2) \cdot h_{k_2} \right) \cdot l_{k_1} \tag{3.4}$$

$$f^{LH(g-1)}(i,j) = \sum_{k_1=0}^{L-1} \left(\sum_{k_2=0}^{L-1} f^{LL(g)}(2i+k_1, 2j+k_2) \cdot l_{k_2} \right) \cdot h_{k_1} \tag{3.5}$$

$$f^{HH(g-1)}(i,j) = \sum_{k_1=0}^{L-1} \left(\sum_{k_2=0}^{L-1} f^{LL(g)}(2i+k_1, 2j+k_2) \cdot h_{k_2} \right) \cdot h_{k_1} \tag{3.6}$$

The computation of the four DWT sub bands (LL, HL, LH and HH) is described by equations (3.3), (3.4), (3.5) and (3.6), respectively, where $f^{Z(g)}(i,j)$ represents the coefficients for sub-

band Z with resolution of g according to Bogush [7]. l_{k_1}, l_{k_2}, l_{h_1} and l_{h_2} are the low pass filter and highpass filter coefficients respectively. L is the horizontal and vertical dimension of the matrix the DWT is applied to.

Because the DWT is calculated by not only examining a pixel's intensity but also that of its neighbors, the results contain a spatial data component organized by the three sub-bands. The HL (horizontal) sub-band will respond greater to intensity gradients between a pixel and the pixel to its right, the LH (vertical) sub-band will respond greater to gradients between a pixel and the one below it and the HH (diagonal) sub-band will respond greater to gradients to the lower right.

3.2.4 Pyramid

After decomposing a tile into its component sub-bands, the LL sub-band can be further decomposed into yet another four coefficient sub-bands, divided again by 2 along both axes yielding a new octave of sub-bands containing 128×128 coefficients. This process can be repeated, forming additional four sub-band pyramids until the sub-bands are 1×1.

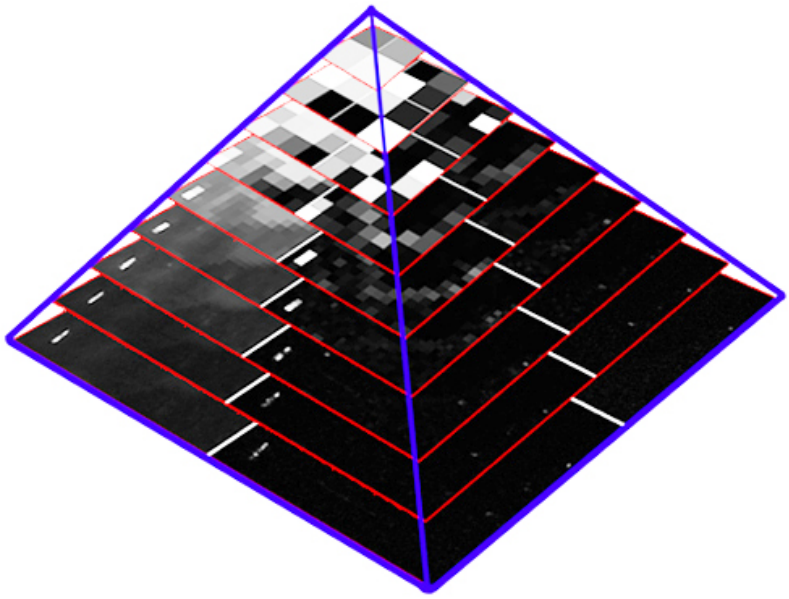

Figure 3.5: A pyramid with 9 octaves, each containing 4 wavelet sub-bands.

The pyramid height, or highest number of octaves to be added to the pyramid, is configurable. Adding octaves to the pyramid generally results in more detections, but requires more computation time and memory. Furthermore, too many octaves within the pyramid will cause too many non-ship pixels to be detected.

Because each octave is repeatedly divided, each successive octave contains only one quarter the coefficients as its precursor. To properly calculate a feature vector that samples the appropriate DWT coefficients from each pyramid octave, multiple coordinate transforms must be performed for every pixel of data within the source image. While mathematically straightforward, performing the transform can be prohibitively slow as there are often millions of pixels in the satellite image, with several octaves, each with three sub-bands to calculate per pixel.

OASIC uses a shortcut that yields the exact same results yet performs far faster. The shortcut is to scale each octave to match the size of the largest octave at 256×256. Once all octaves are the same size, they can be combined into a three dimensional matrix, and the feature vectors can be used to create a larger wavelet pyramid vector with no additional floating point operations as shown in Figure 3.6. The speed boost comes at the cost of memory as each scaled pyramid octave consume 2^{2n} times as much memory where n is the octave.

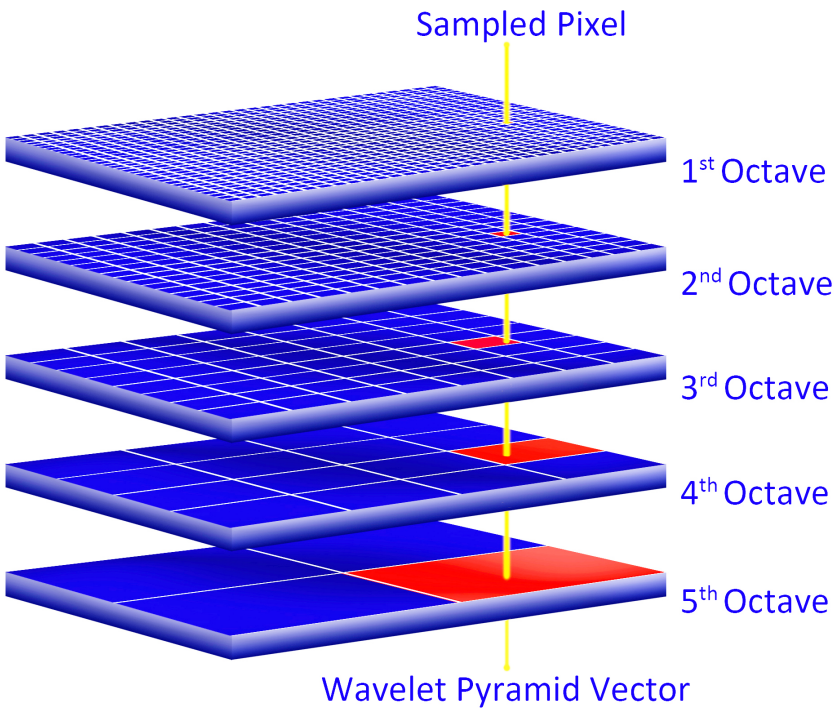

Figure 3.6: An example of how a single coefficient of a wavelet sub-band is aligned to its four higher octaves and the appropriate coefficients are retrieved and combined into a feature vector.

When scaling wavelet pyramid octaves, the scaled image may be interpolated using nearest neighbor with no distortion as the octaves are always interpolated by integer factors. However, OASIC offers the ability to interpolate the pyramid octaves using bilinear, trilinear or bicubic filters. True positive (TP) pixels are correctly identified pixels, while false positive (FP) pixels

are non-ship pixels erroneously identified as ship-pixels. Using these interpolation methods has the effect of improving TP pixel detection rates significantly over nearest neighbor while raising the FP pixel rates by a much smaller rate. The results of using bicubic interpolation will be shown in Chapter 5.

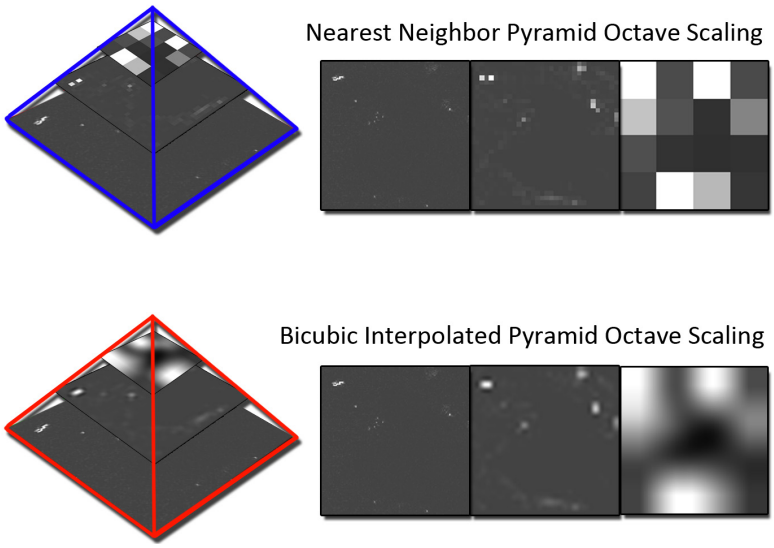

Figure 3.7: The 3-octave is scaled by nearest neighbor (top) or bicubic filter (bottom).

3.3 Classification

The heart of OASIC is its ability to properly identify each pixel of an image as belonging to either a ship or the ocean. Unlike many ocean vessel detection schemes, OASIC makes no attempt to recover any additional information about the ship such as its speed, course, type, or identity. The goal of OASIC's classification is to determine the vessel's existence and location within the image for the purpose of selective compression only.

Therefore, OASIC is tolerant of much higher false positive pixel rates than other detectors. The emphasis is on maximizing true positive pixels at the expense of detecting false negative pixels (ocean features erroneously detected as ships, such as wave crests).

3.3.1 Support Vector Machine

The SVM was chosen as OASIC's classifier due to its excellent operating characteristics when training and predicting between only two labels: ships and ocean.

Inputs to the SVM are provided by the wavelet pyramid and its octaves. The coefficients of multiple octaves spanning the pyramid are retrieved and are then combined into the Wavelet

Pyramid Vector for each pixel within the image. The LH, HL and HH sub-band values from each octave are combined in the order defined in Equation (3.7) where m is the number of octaves to be sampled, and Equation (3.8) where **S** is the Wavelet Pyramid Vector with n pixel samples. The **S** vector, once calculated, will be passed onto the SVM. Note: The LL sub-band is not part of the **S** vector.

$$\mathbf{aV_n} = < LH_1, LH_2 ... LH_m >, \mathbf{aH_n} = < HL_1, HL_2 ... HL_m >, \mathbf{aD_n} = < HH_1, HH_2 ... HH_m > \quad (3.7)$$

$$\mathbf{S} = < aV_0, aH_0, aD_0, aV_1, aH_1, aD_1 ... aV_n, aH_n, aD_n > \quad (3.8)$$

3.3.2 Training

OASIC uses a single 512×512 pixel representative image for training. This image contains clouds, large and small vessels, cloud shadows and some wave crests. Once feature extraction has been performed, the SVM trains on this image's pyramid. Paired with the 512×512 pixel training image is a matrix of *ground truth* labels of the same dimensions called an Annotation Label Matrix, labeling each individual pixel as either ship or non-ship.

3.3.3 Prediction

Each 512×512 tile from the source image is supplied to the feature extractor which performs the exact same processes on this image as the training image. Note that due to the 2^n tile dimensions, no pyramid octave can ever overlap adjacent tiles and no seams or artifacts are produced by tiling due to borders between adjacent tiles.

During prediction, the SVM fills a label matrix for each tile which is combined to form an matrix of predicted labels of the same dimensions as the original image. From this matrix, the ships can be extracted from the background in a process called Layer Segmentation.

3.4 Layer Segmentation

Once the matrix of predicted labels is calculated, the source image can be segmented into two distinct regions: the foreground and background layers. The foreground layer contains all detected ship-pixels while the background contains all other pixels. Layer segmentation permits selective compression as both layers can be compressed independently.

3.4.1 Two-Layer Method

The most straightforward method to take advantage of the two segmented layers is to compress the foreground with a lossless fidelity, allowing the pure black pixels to serve as transparent pixels, or including a 1-bit transparency mask which itself can be efficiently compressed. The background is compressed with a low quality lossy compression. The two files are combined in the same container file.

This method makes no attempt to take advantage of the known location of the foreground objects within the image. Rather, the Two-Layer Method relies on foreground compressor to efficiently compress the layer by taking advantage of the long runs of zeros present between objects in the sparsely populated foreground layer.

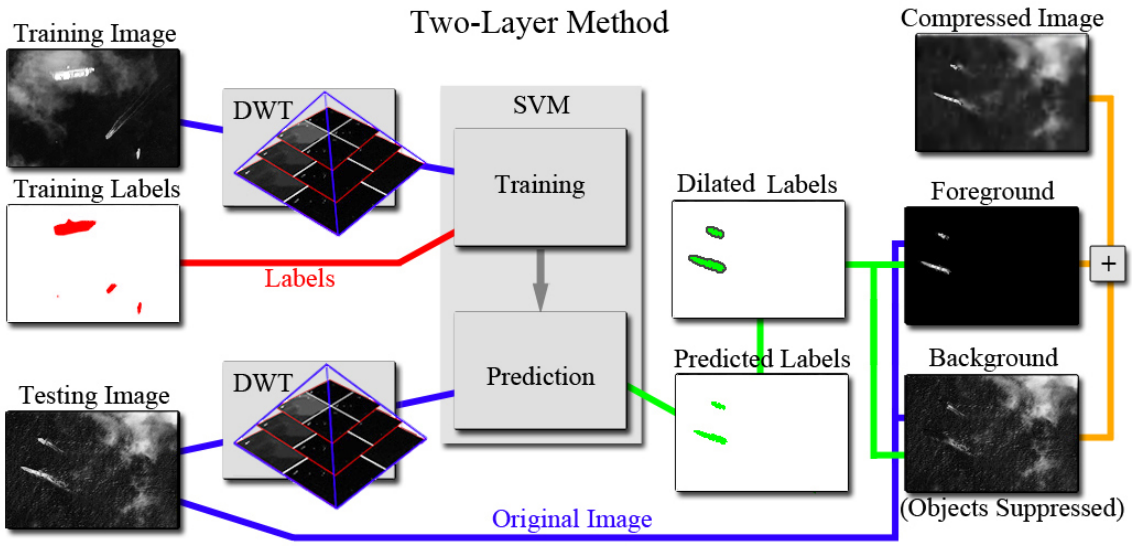

Figure 3.8: A block diagram displaying the Two-Layer Method flow.

3.4.2 Bounding Rectangle Method

The Bounding Rectangle method takes advantage of the separation of foreground objects from the background ocean but further breaks down the foreground to eliminate the empty space between detected clusters of pixels. The foreground layer is decomposed into rectangles by using a quadtree algorithm.

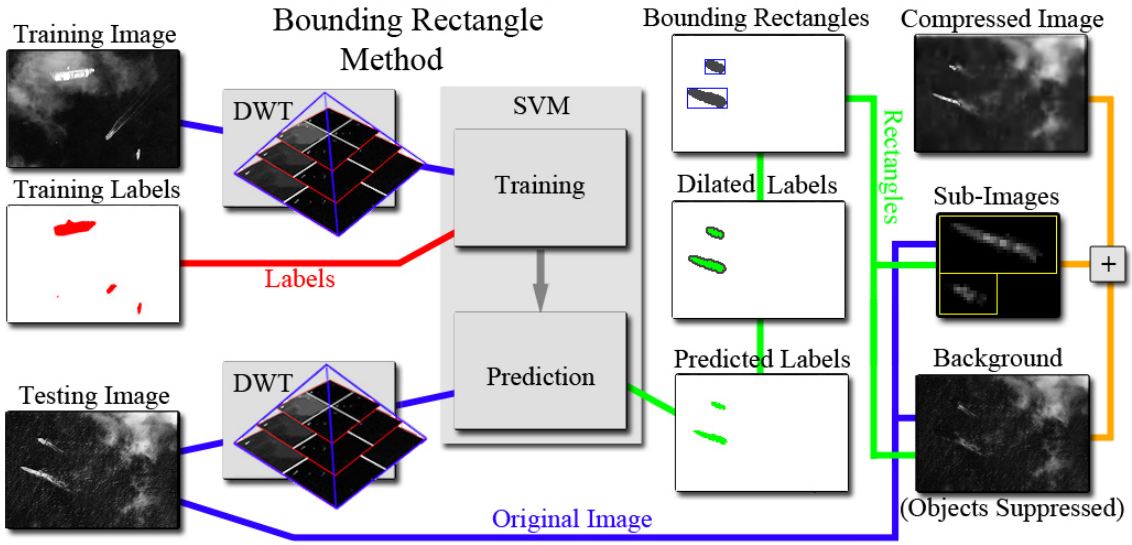

Figure 3.9: A block diagram displaying the Bounding Rectangle Method flow.

3.4.3 Decomposition by Quadtree

The purpose of the quadtree is to provide a list of coordinates that define axis-aligned bounding rectangles that enclose ship clusters. OASIC's implementation of the quadtree does not create a quadtree data structure. The quadtree functions by subdividing an image into four quadrants without cutting any objects into pieces. The two axis-aligned dividing lines for the new division start at the center and are perpendicular to each other. If the dividing lines fall on a non-zero pixel value, two temporary lines are created along the same axis and shift along the dividing line's perpendicular axis in both directions until one of the temporary lines no longer falls on a non-zero pixel or reaches the border of the image. The first temporary line to find a row or column with no pixels will become the new location for that dividing line. Once both horizontal and vertical dividing lines are established, the image is divided into four smaller images and each subdivision is recursively subdivided further. Once the temporary lines cannot avoid non-zero rows, an image can no longer be divided. The result is that all objects or clusters of objects are enclosed by axis-aligned rectangles to the closest extent possible. The enclosing axis-aligned rectangles are illustrated in Figure 3.10 enclosing ships.

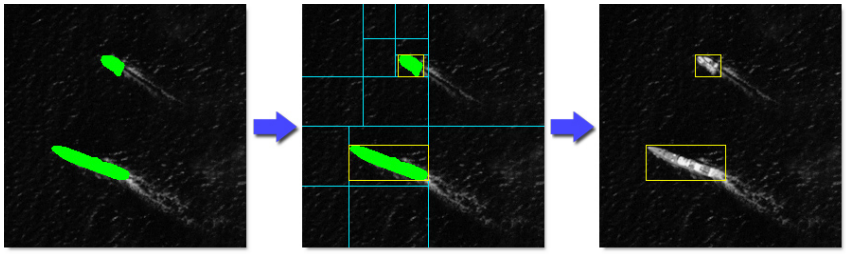

Figure 3.10: Two objects are bounded by a quadtree algorithm.

The implementation of the quadtree used is designed specifically for OASIC to be very fast, even when working with very large images as long as they are sparsely populated. Once the foreground has been decomposed into a number of varying sized rectangles, all empty rectangles are deleted, and each remaining rectangle is left enclosing one or more groups of foreground pixels. For each rectangle, the upper-left coordinates are stored along with the dimensions.

Each object bearing rectangle, henceforth referred to as an sub-image, must still be compressed. Early experimentation showed that compressing each individual sub-image quickly grew costly due to the objects being too small for entropy based compressors to be efficient. Furthermore, each compressed sub-image contained its own header, sometimes larger than the sub-image itself. To address the inefficiencies of individual sub-image compression, an efficient rectangle packing algorithm is employed to combine all sub-images into a single foreground composite rectangle.

3.4.4 Efficient Rectangle Packing

Efficient rectangle packing permits the merging of all foreground sub-images into one large rectangular image with minimal gaps.

Using a derivative of the method described by Korf [23], any arbitrary number of irregularly shaped rectangular, axis-aligned sub-images can be packed quickly and efficiently. Figure 3.11 displays a packed rectangle with the largest vessels placed first, and the smaller vessels used to fill in any gaps. Figure 3.12 is the same algorithm used on an image containing nearly 600 detected vessels. Once assembled, the composite foreground rectangle is then compressed as the new pseudo-foreground along with the coordinates of the sub-images within both the packed rectangle and the foreground image. The sub-image dimensions are also stored. Each sub-image requires 12 bytes of overhead to store its coordinates and dimensions.

The shape of the rectangle is as close to a square as possible to equalize the number of pixels in both the horizontal and vertical axes. Many lossless compression algorithms such as JPEG2000 take advantage of spatial repetition. This method, by virtue of the packing algorithm, maximizes this repetition along both horizontal and vertical axes and permits better lossless compression. Early experimentation indicated the improvement in efficiency to be relatively minor, especially with large images.

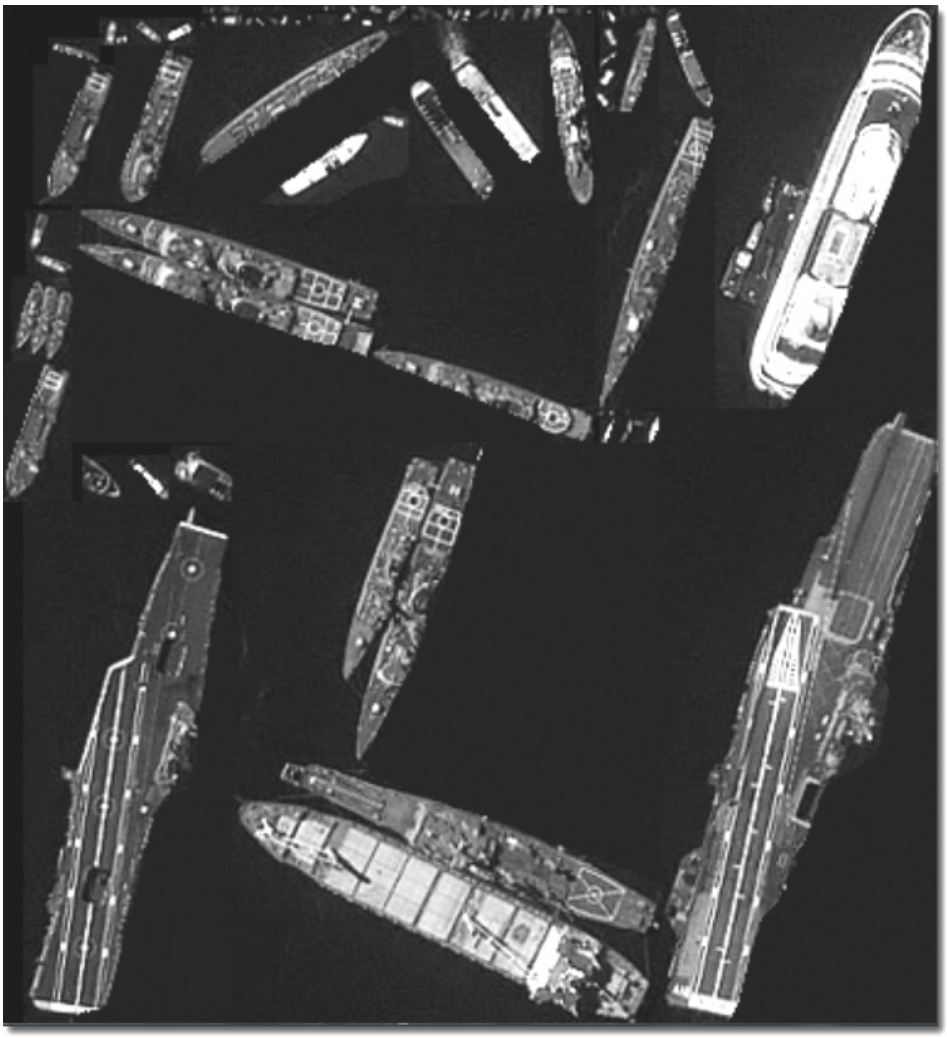

Figure 3.11: Example of efficient packing of a few sub-images.

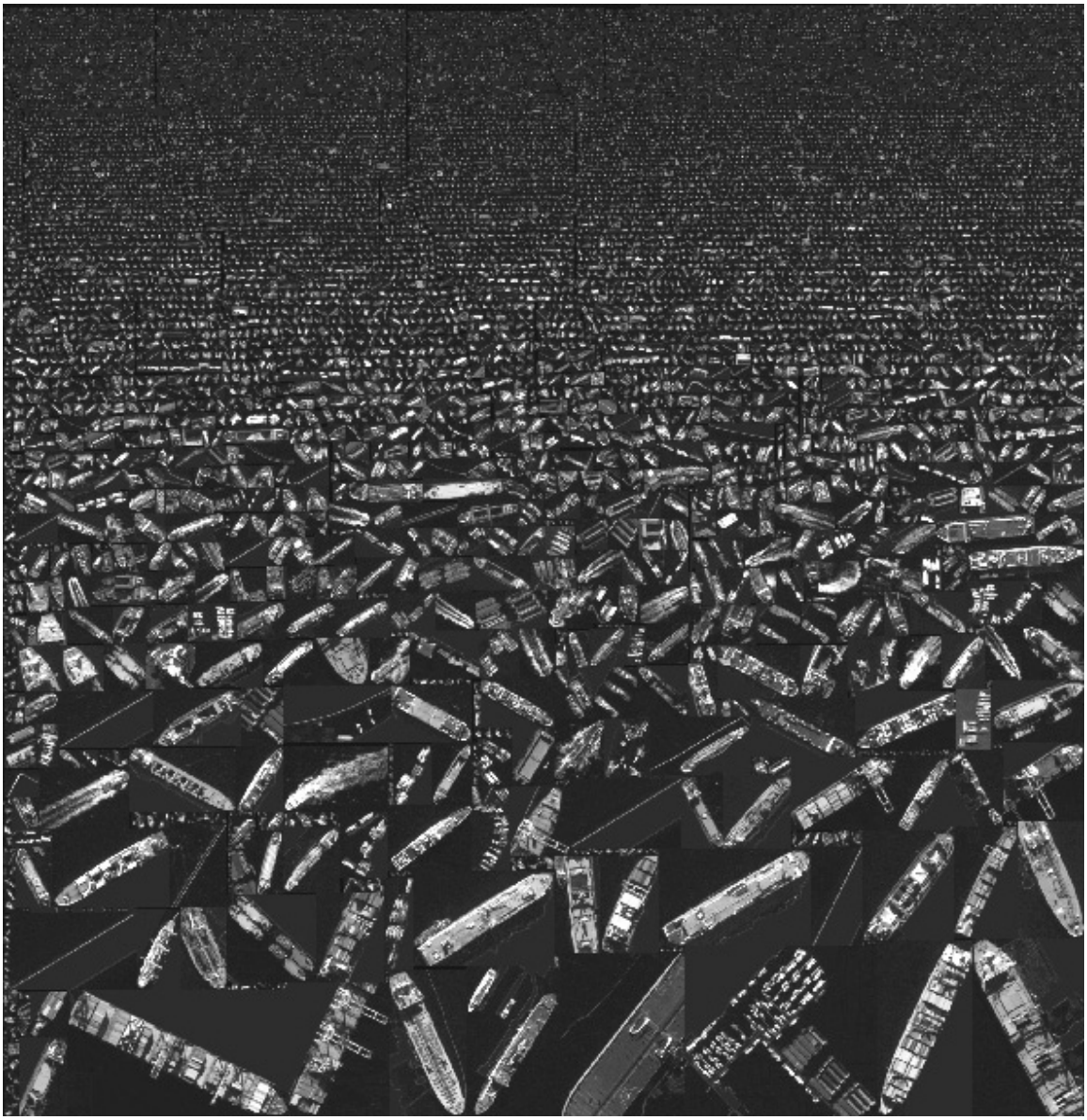

Figure 3.12: Example of efficient packing of nearly 600 sub-images.

3.4.5 Object Dilation

Detecting large vessels with features extracted with the Discrete Wavelet Transform can be problematic as their internal areas often have little contrast and tend to not stimulate a large enough response from the DWT for the Support Vector Machine to recognize them as ship-pixels. This shortfall leaves large gaps within the detected area of a vessel as shown in Figure 3.13(a).

A simple and effective solution is to designate a radius around each detected pixel as ship-pixels, even if they were not originally classified as such. The morphological preprocessing operation used achieve this result is called dilation [27, p. 490] and is computationally inexpensive. Dilation causes more of the surrounding ocean to be preserved, can fill hollow spaces and close gaps within larger objects. Dilation, however, increases the number of false positive pixels.

3.4.6 Object Preprocessing Solutions

More dilation generally means more of the vessel is detected. However, with 4 pixel dilation, or even 8 pixel dilation, gaps still exist for the largest ships as shown in Figure 3.13(b) and (c) respectively. OASIC supports two additional preprocessing solutions that attempt to better enclose the vessel using the ship-pixels that are detected.

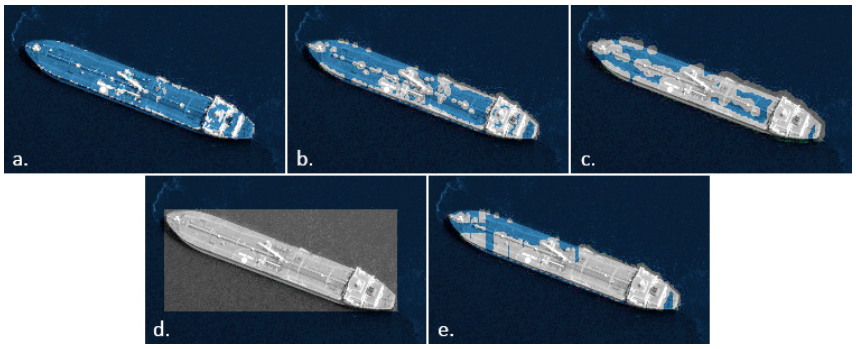

Figure 3.13: a. No dilation b. 4-pixel dilation c. 8-pixel dilation d. Solid Rectangle e. Filled Object The foreground is indicated by the lighter gray and the background by darker blue.

Solid Rectangle

A simple and effective solution is to simply enclose the entire cluster of pixels within a bounding rectangle as shown in Figure 3.13(d). The entire bounding rectangle of the sub-image is captured and added to the foreground.

Filled Object

The Filled Object method is a preprocessing operation designed for use with OASIC. It bears some resemblance to Smart Snakes by Cootes [28] but differs in implementation. Filled Object uses orthogonal rays cast from the top, bottom, left and right edges of the sub-image to fill in the object. The rays terminate once encountering a ship pixel. When all rays have been terminated, any pixels not traversed by a ray are classified as ship pixels as shown in Figure 3.13(e). This method works best if some dilation is used first.

3.5 Selective Compression

Most compression schemes work by taking advantage of the inherent redundancy found in an image. OASIC, however, takes advantage of the relative sparsity of ships present within the ocean. Only the detected ships are preserved by the lossless compression of the foreground, while the ocean is distorted by the extremely lossy background compression.

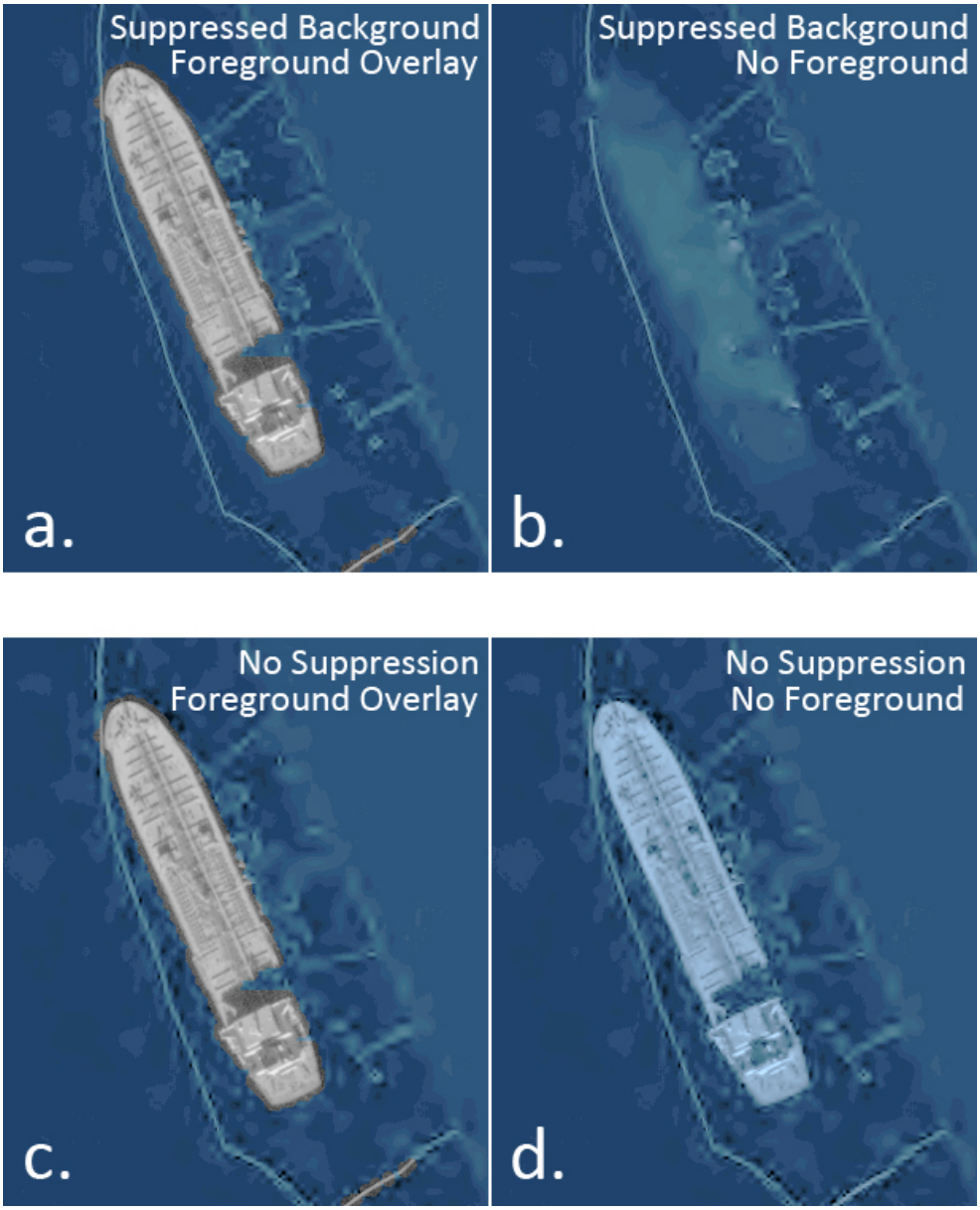

Figure 3.14: Images with suppression (top) suffer from less noise than those without suppression (bottom) where ringing artifacts are more prominent. Images with foreground (gray) disabled (right) shows that suppression removes some distortion from the background (blue).

3.5.1 Object Suppression

Although the background is already compressed with a lossy algorithm, configured with a low fidelity, and achieves a tremendous reduction in size, there is one further optimization: The objects that have been detected and compressed as the foreground are still present in the background. By removing them, less information needs to be stored to represent them since they are already stored in the foreground at a higher fidelity. This step also eliminates the occurrence of ringing artifacts around the object that extend beyond the original objects boundaries as shown in Figures 3.14(a) and (c).

The detected objects are removed from the background layer by replacing their pixels with a content-aware gradient of pixel shades as shown in Figure 3.14(b). Suppression of background objects not only improves compression but improves the fidelity of partially detected ships as undetected internal areas are not corrupted by the ringing artifacts caused by the unnecessary compression of the ship in the background. This corruption is shown in Figure 3.14(c) and (d).

3.5.2 JPEG

JPEG, defined by ITU-T T.81 and ISO/IEC 10918-1 [29] is a lossy compression format with an adjustable fidelity that encodes an image with a discrete cosine transform (DCT), quantizing the products and achieving an impressive compression ratio. OASIC evaluates this method's performance as a background compressor.

3.5.3 JPEG 2000

JPEG2000 is defined by ITU-T T.800 and ISO/IEC 15444-1 [20] and functions in both lossy and lossless modes.

Lossy JPEG2000 encodes an image in much the same way as the detector stage of OASIC, in that it decomposes an image into a pyramid using the Discrete Wavelet Transform (DWT). Like JPEG, it too, has a configurable fidelity. Due to its superior method of storing the coefficient products of the DWT over regular JPEG, JPEG2000 achieves a much better compression ratio with far better quality.

OASIC evaluates both of this method's modes, using lossless for its foreground compression and lossy for its background compression. The actual file container format used by both OASIC and for comparison with OASIC is the JP2 minimal JPEG2000 format [20].

3.5.4 PNG

The PNG standard is define by ISO/IEC 15948 [30]. This is another lossless file compression format that functions very similarly to the GIF file format it was intended to replace. OASIC evaluates this method as well for use in compressing its foreground.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Experimentation

The evaluation of OASIC's algorithms is a multi-part problem. A large testing set of annotated oceanic satellite imagery must be evaluated for detection with different configurations and compression compared to both lossy and lossless algorithms.

4.1 Equipment and Software

Computer Specs

All testing is performed on one system with an AMD Athelon^{TM} 64 X2 Dual Core CPU at 2.6GHz with 4.00Gb of RAM running Windows 7 64-bit Home Premium.

Implementation

OASIC's algorithms are written in Mathworks MATLAB (R2012b) due to the ease of processing large amounts of data in matrix form. MATLAB also natively provides support for configuring, saving and loading exotic image formats such as lossless JPEG and JPEG2000. The only non-standard MATLAB toolbox used is the LibSVM for the Support Vector Machine.

4.2 Testing Performance

The performance of OASIC is evaluated at different stages: The Detection Stage's Wavelet Pyramid configuration and dilation/preprocessing options are tested and the Compression Stage's performance is compared to both JPEG2000's lossless and lossy modes.

4.2.1 Image Annotation

Just as with the training image discussed in Chapter 3, all satellite images to be tested are first annotated. Because OASIC is foremost a compression algorithm, and not a detection algorithm, vessels are not distinguished from each other in the Annotation Label Matrix supplied with each satellite image. Annotation is done on a per-pixel level, with a 1 corresponding to a ship-pixel in the source image and a 0 corresponding to a non-ship pixel. Red pixels represent ship pixels in Figure 4.2.

4.2.2 Detection Evaluation

Judging whether a ship has been successfully detected is not necessarily a straightforward problem. Simple methods such as simply enclosing both true and detected ships in a bounding box and measuring their area of overlap are quick but dependent on the orientation of the ships. Vessels at diagonal orientation do not fit efficiently within rectangles and can impact testing accuracy.

OASIC uses a per-pixel evaluation by comparing every detected pixel to every annotated pixel and calculating the percentage of detected pixels for the entire image and for ship clusters. This analysis can be efficiently performed by using Equation 4.1 to determine the percentage of detection per the entire image. Note that this evaluation method is more precise and hence stricter than the rectangle overlap method or other common methods used to evaluate detectors.

Once an OASIC compressed image is uncompressed, a copy of the original predicted label matrix is derived from its foreground layer D. To convert these values back into binary values, the mathematical sign is used. The $sgn(D)$ can be thought of as a bit mask, and when applied to the Annotation Label Matrix R using the logical AND operator, the only pixels remaining are true positive pixels. By converting these pixels into binary values using the mathematical sign function, summing them and then dividing the sum by the total ship pixels, the True Positive Pixel detection rate (T_P) is calculated. In the equation, m and n correspond to the image dimensions in pixels, and i and j are their indices.

$$T_P = \left(\frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} sgn(D[i,j]) \wedge R[i,j]}{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} R[i,j]} \right) * 100 \tag{4.1}$$

The result is the True Positive Pixel detection rate for the entire image. As mentioned before, the Annotation Label Matrix does not distinguish individual ships from one another. Therefore, in order to gather *per-ship cluster* statistics, the Annotation Label Matrix must be broken up into localized ship clusters (Shown red in Figure 4.1). Fortunately, the quadtree algorithm discussed in Chapter 3 is perfect for this task.

Once the percentage of ship pixels within a localized ship cluster is calculated, the performance of the detector can be further broken down: Any percentage below 50% is considered a failure to detect the ship. The number of detections at 50%, 75% and full 100% are calculated and graphed.

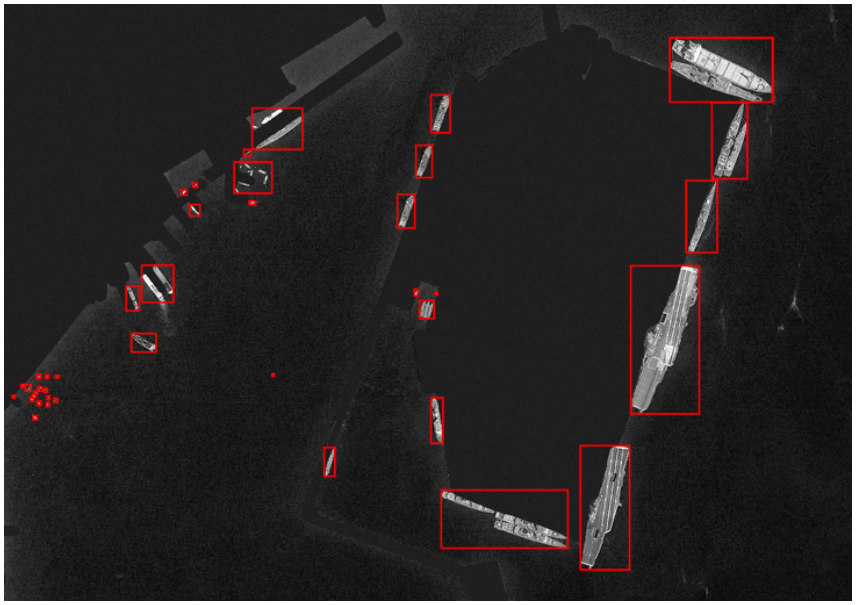

Figure 4.1: Image with the ship clusters enclosed in rectangles (red).

4.2.3 Detector Configuration

Pixel dilation, pyramid octaves and object preprocessing options are all configurable and all effect detection efficiency.

Pyramid performance with different octaves are tested to determine the best number of octaves to use for detection. All octaves beyond the first are scaled using nearest-neighbor interpolation, but the results of using bicubic interpolation are tested as well.

Five preprocessing configurations are analyzed: No dilation, dilation with a 4-pixel radius, and dilation with an 8-pixel radius. The Solid Rectangle and Filled Object (using 4-pixel dilation) methods are also tested.

The result of the detection experiments are presented as a Receiver Operating Characteristic (ROC) curves which are well suited to spot trends in the relationship between True Positive Pixels and False Positive Pixels. ROC curves will be produced for the different pyramid configurations, different dilation options, solid rectangle and filled object methods.

4.2.4 Comparing Compression Ratios

The compression ratio (C/R) is defined as the original uncompressed image size divided by the compressed file size as shown in Equation 4.2. The value of a compression ratio R is expressed R:1 (pronounced R to 1.)

$$R = \frac{U}{C} \tag{4.2}$$

To calculate the compressed file sizes, both PNG and lossless JPEG2000 are considered.

The lossless PNG format produces larger files than the lossless JPEG2000 algorithm in all of the 10 large satellite images tested. (An average of 17% larger.) Similarly, the lossy JPEG format introduce approximately 8% more noise to an image than its lossy JPEG2000 counterpart for the same file size. For these reasons, comparisons are made only using lossless JPEG2000 for the foreground layer, and lossy JPEG2000 for the background layer. Comparing OASIC to JPEG2000 in lossless mode is done by simply calculating the compression ratios of the two and using this comparison as a measure of OASIC's performance.

To evaluate OASIC's performance in ocean imagery compression, the testing set is compressed both in OASIC's OAI format, and JPEG2000's minimal JP2 format. Because OASIC gains its efficiency by taking advantage of the relative sparsity of ships at sea compared to the ocean and masked terrain, small image chips will perform less favorably when compared to full scale satellite images. For this reason, full sized satellite images are evaluated to test performance by compressing them with the OASIC algorithm with the optimum pyramid configuration, dilation and preprocessing options.

The images are compressed within five kilobytes of their OASIC counterpart's file size with the minimal lossy JPEG2000 format (JP2). The noise produced by both algorithm's lossy compression is evaluated to compare fidelity.

OASIC's lossy background layer's compression ratio is fixed at 500:1. Therefore, the theoretical maximum compression ratio for any OASIC file is 1/500th the uncompressed size. (With no ships present in this extreme case.)

4.2.5 Comparing Fidelity Loss

All lossy compression algorithms introduce noise, however, OASIC and JPEG2000 distribute their noise in completely different fashions. This experiment will confirm that OASIC introduces less errors to the ship pixels than JPEG2000 does for the same file size.

JPEG2000's lossy mode cannot be compared by compression ratio because its compression ratio is dependent on its fidelity setting. In order for such a comparison, both OASIC and the

lossy JPEG2000 must have a similar level of fidelity for such a comparison to be meaningful. This is problematic because it is difficult to match fidelity, or level of noise, between the two compression algorithms. However, JPEG2000's target file size can be precisely set (within approximately 5 kilobytes), allowing for lossy JPEG2000 compressed files to match compression ratios with their OASIC compressed counterparts. The errors (inverse of fidelity) for both files are then calculated and compared to measure the performance of both algorithms.

To evaluate the overall error introduced by the lossy compression, the PSNR (Peak Signal to Noise Ratio) must be calculated. This is done by first calculating the MSE (Mean Square Error) from the original image I and the lossy compressed image K as shown in Equation 4.3 where m and n are the dimensions of the image. Once the MSE M is obtained, the PSNR P can be calculated using Equation 4.4 with b as the common bit depth of the images. (All images are 8-bit for this experiment.) The PSNR is in decibels (dB), with higher values indicating higher fidelity of the lossy image, and the lower values indicate worse fidelity. An infinite PSNR indicates a lossless image.

$$M = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{4.3}$$

$$P = 20 \cdot log_{10} \left(\frac{2^b - 1}{\sqrt{M}} \right) dB \tag{4.4}$$

The PSNR metric is most useful when comparing the exact same regions of the same images, so the entire image is evaluated, and the PSNRs of the individual ships are summed and evaluated separately.

4.2.6 Image Set

The images used for training the Support Vector Machine are crucial to the performance of OASIC. Training images should ideally match the expected circumstances of the image to be compressed, if known. Poor weather should warrant a training image with more cloud cover, while rough seas should necessitate a training image with the presence of white caps. (Waves crests that appear white from above.) If the user or satellite does not have any knowledge of the weather or sea state before hand, a generic image can be used to train with such as indicated in Figure 4.2.

Training Images

OASIC can be configured to use any training image, however, the image must be 512×512. Only one training image was used for all experiments. The image used is depicted in Figure 4.2.

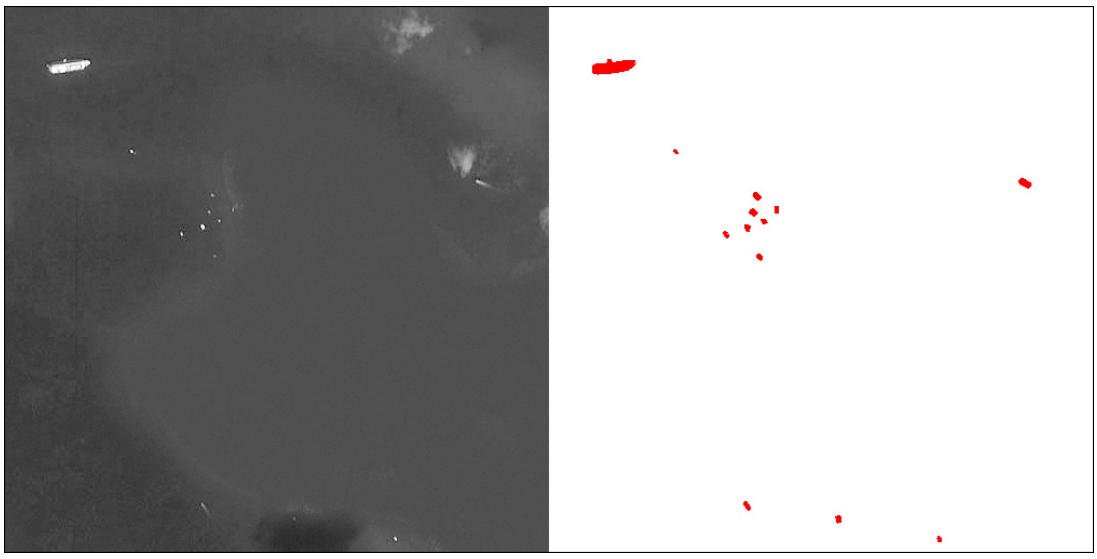

Figure 4.2: A training image with its associated labels (red) showing a mix of large and small vessels and clouds.

Testing Images

The image test set is comprised of several color images from around the world including heavily trafficked ports, open ocean, extreme cloud cover, and sea states from a calm 0 to a tumultuous 7 on the Beaufort Scale. All images were obtained from commercial satellites and provided by Space and Naval Warfare Systems (SPAWAR).

All images are subsequently converted to panchromatic for testing. For the compression experiments, 10 full sized (221.5 to 775.5 megapixels) images are used.

For ship detection experiments, 25 image chips (1 to 16.8 megapixels) are extracted. This step is done for speed considerations yet will have no effect on accuracy so long as the 25 images are sufficiently representative of the environments found in the 10 images.

CHAPTER 5:
Results

The performance of OASIC is analyzed according to the criteria described in Chapter 4, addressing first ship detection accuracy, then lossless and lossy compression.

5.1 Ship Detection

Preliminary analysis has indicated that an optimal pyramid configuration is useful for discerning waves and clouds from vessels as they otherwise may confuse the SVM. Determining such a configuration is the first experiment. Once the best pyramid configuration is established, all subsequent experiments use this configuration.

5.1.1 Optimum Pyramid Configuration

Various pyramid configurations are tested on an annotated image containing clouds, masked terrain and ships of varying sizes and orientations. For the pyramid configuration tests, no pixel dilation or any other preprocessing method is applied to its predicted label matrix. The independent variable is the number of pyramid octaves while the dependent variables are numbers of true positives pixels and false positive pixels (ocean pixels misidentified as ship pixels). The results appear in Figure 5.1. This experiment's results indicate that a three octave pyramid is the most accurate, agreeing with previous work by Huang [8], Kiely [9] and Zhu [10].

As described in depth in Chapter 3, scaling each pyramid octave to match the dimensions of the largest octave provides a speed boost because complicated coordinate transforms are no longer needed. Normally, Nearest Neighbor interpolation is used when scaling octaves to precisely emulate the slower coordinate transform that it replaces, but bicubic interpolation can be used instead as shown in Figure 3.7. Repeating the experiment with this method yields an additional 10% boost to accuracy as shown in Figure 5.2.

A 3-octave pyramid scaled with bicubic interpolation is used in the detection stage for all subsequent experiments.

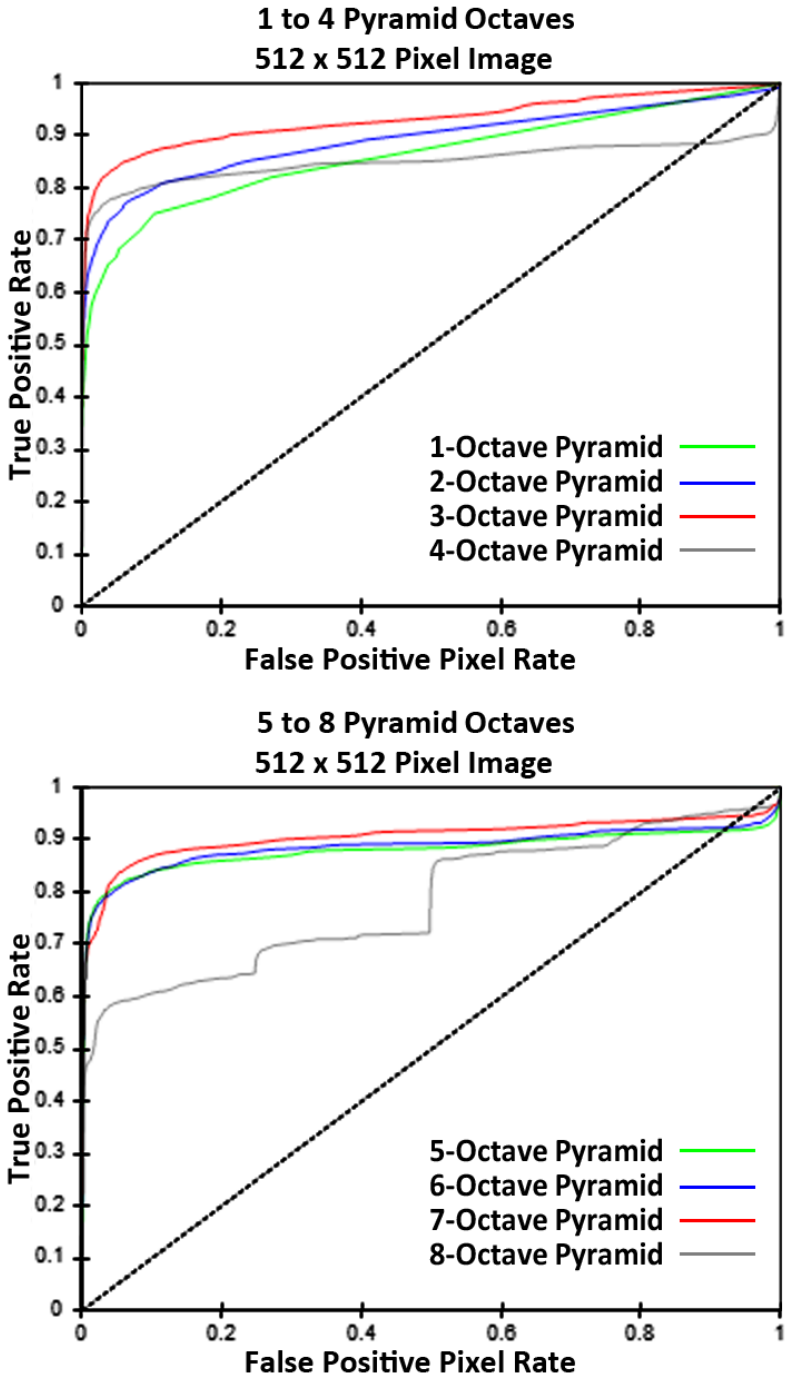

Figure 5.1: With no pixel dilation or octave interpolation, eight pyramid octave combinations are tested. The 3-octave pyramid performs the best.

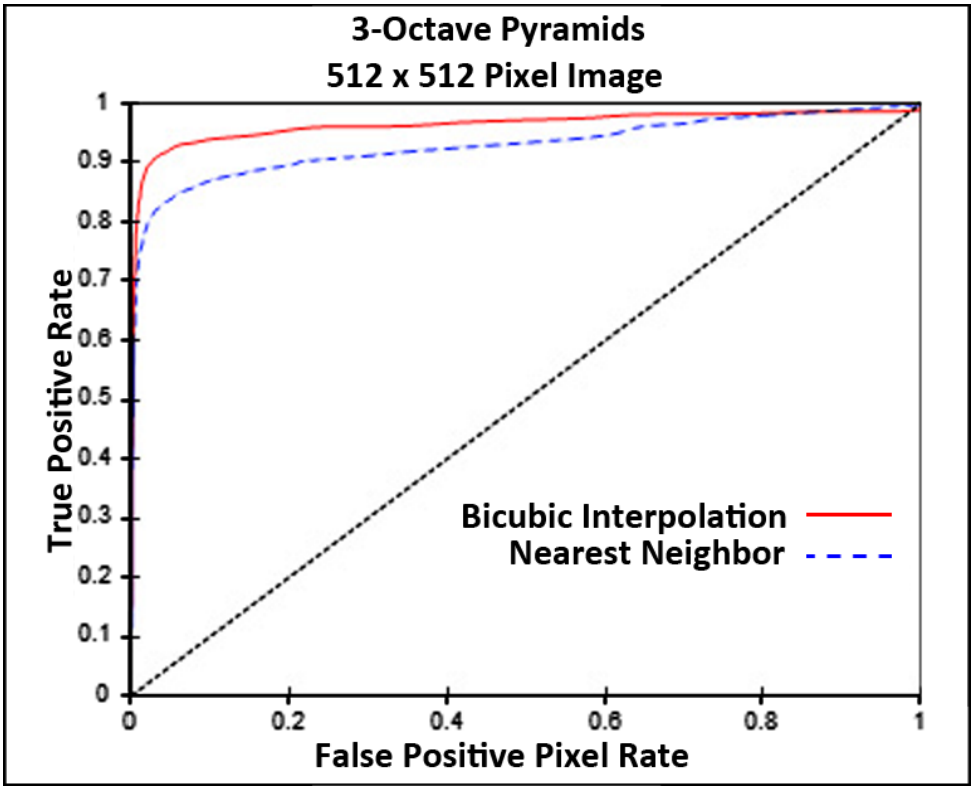

Figure 5.2: The 3-octave bicubic interpolated pyramid (solid line) provides better performance than the standard 3-octave pyramid (dotted line).

5.1.2 Preprocessing Options

This experiment tested 25 images containing 444 ship clusters of varying sizes and orientations in a wide variety of environments. The tests were done with no dilation, 4-pixel dilation, 8-pixel dilation, Solid Rectangle and Filled Object with the results shown in Figures 5.3, 5.4 and 5.5. The ship clusters detection rates are plotted at 50% or greater, 75% or greater and 100% detection intervals. The raw pixel rates are measured and plotted on the same graph as well. The independent variable in this test is the dilation or preprocessing method while the dependent variables are the true positive pixels and false positive pixels.

The optimal pixel dilation radius appears to be 8-pixels, as this method contains the highest number of detections, at only a minor cost to the false positive pixel rate. Pairing 4-pixel dilation with the Filled Object preprocessing method produces results very similar to those produced by the Solid Rectangle method as shown in Figure 5.5. The Filled Object preprocessing method does not appear to perform better than others such as 4-pixel, or 8-pixel dilation.

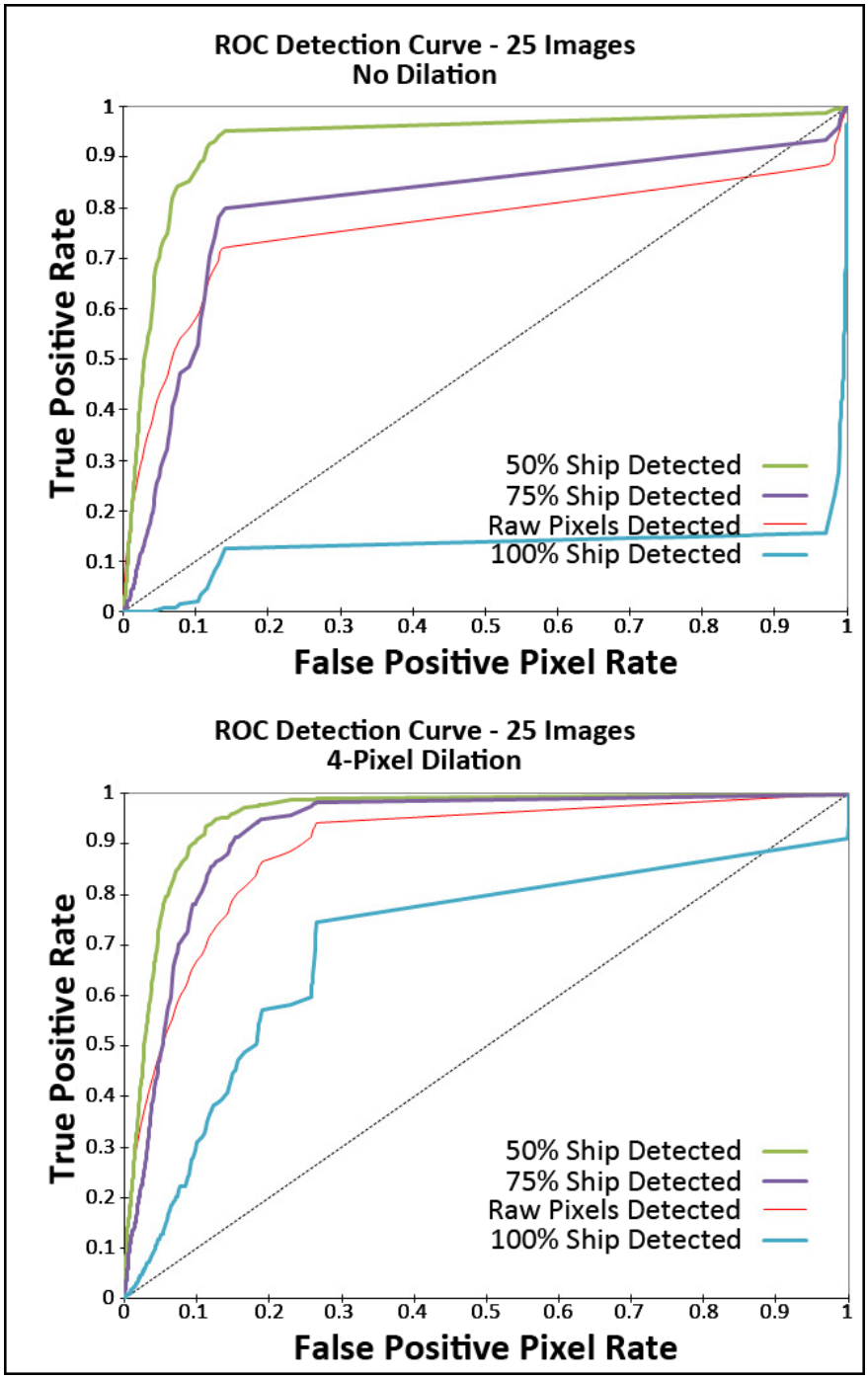

Figure 5.3: Performance with different preprocessing options: No dilation (top) and 4-pixel dilation (bottom)

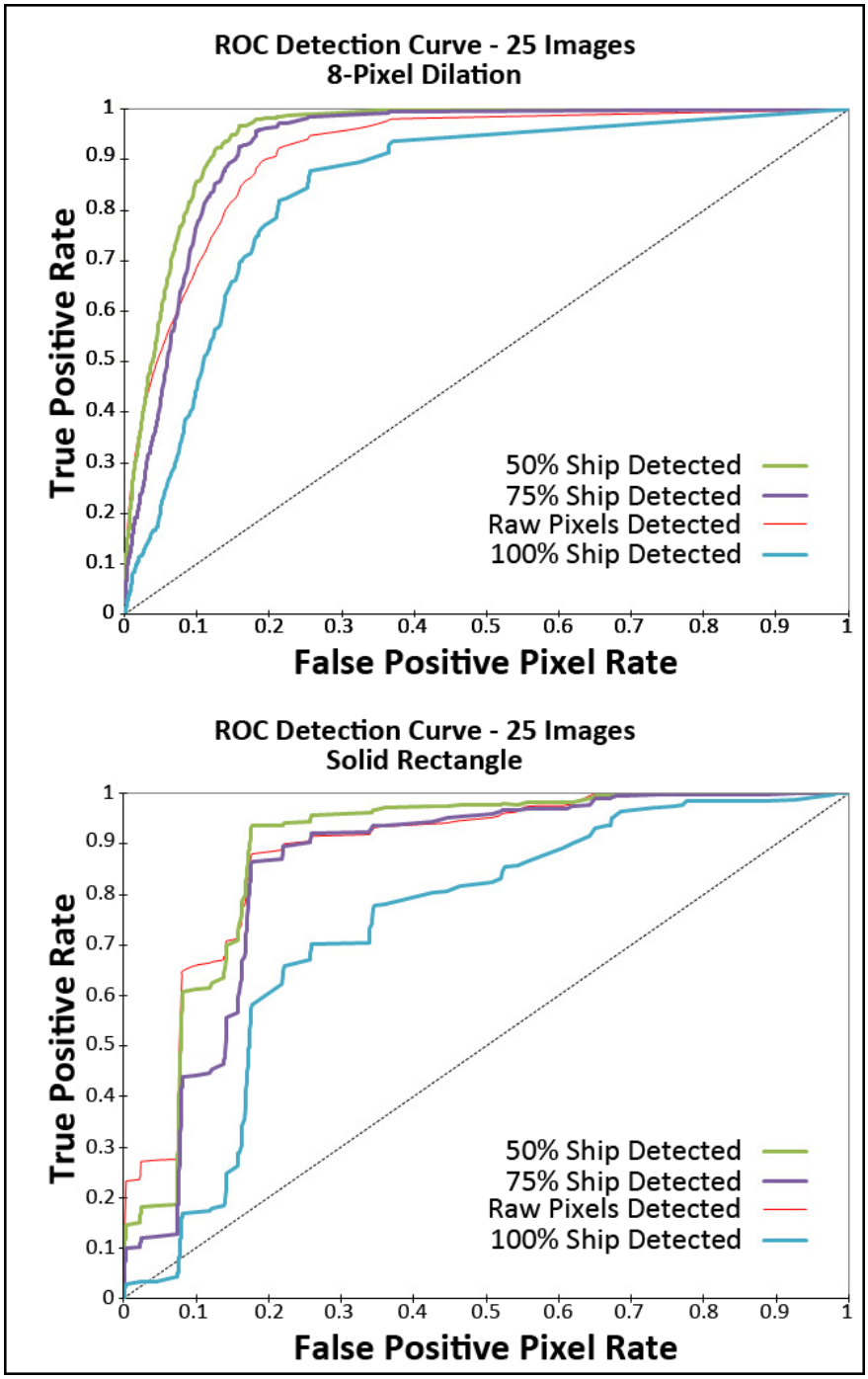

Figure 5.4: Performance with different preprocessing options: 8-pixel dilation(top) Solid Rectangle (bottom)

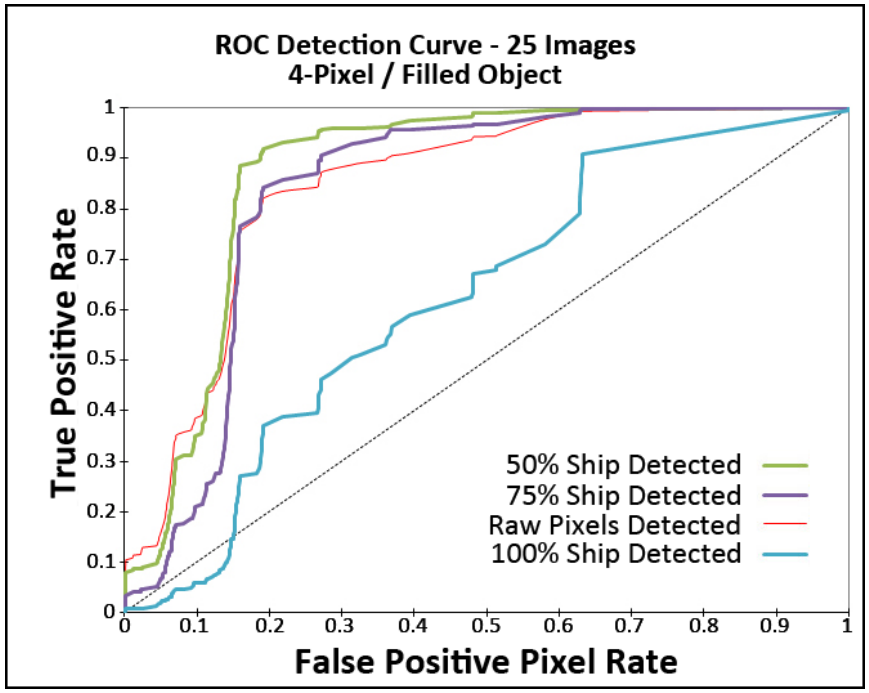

Figure 5.5: Performance of the 4-pixel / Filled Object preprocessing method

5.2 Compression Ratios

The compression ratios of OASIC and lossless JPEG2000 are shown in Figures 5.6 and 5.7 for four of the five tested preprocessing methods. The *No Dilation* method performs poorly and is omitted from these charts. The average compression ratio for all four methods is 113:1, which is 14 times greater than JPEG2000's lossless compression. The 4-pixel dilation method provides the best compression ratio.

The larger vessels tend to contain hollow voids with only their outlines being detected. Dilation fills these voids, improves detection and reduces noise. However, it can undermine compression efficiency by adding pixels around smaller vessels that do not suffer from the hollow void phenomenon. For this reason, 8-pixel dilation performs poorly as the number of pixels filled in is not proportionate to the number of false negative pixels generated. The false negative pixels generated by 8-pixel dilation will cause vessels that are close to each other to be merged under the same ship cluster and cannot be divided by the quadtree when attempting to decompose the foreground, causing more non-ship pixels to be stored in the foreground, undermining the compression ratio. Solid Rectangle and Filled Object both offer better performance compression performance because they discriminate which ships will gain additional additional pixels. Both can fill the voids within larger vessels with a minimal impact to smaller vessels.

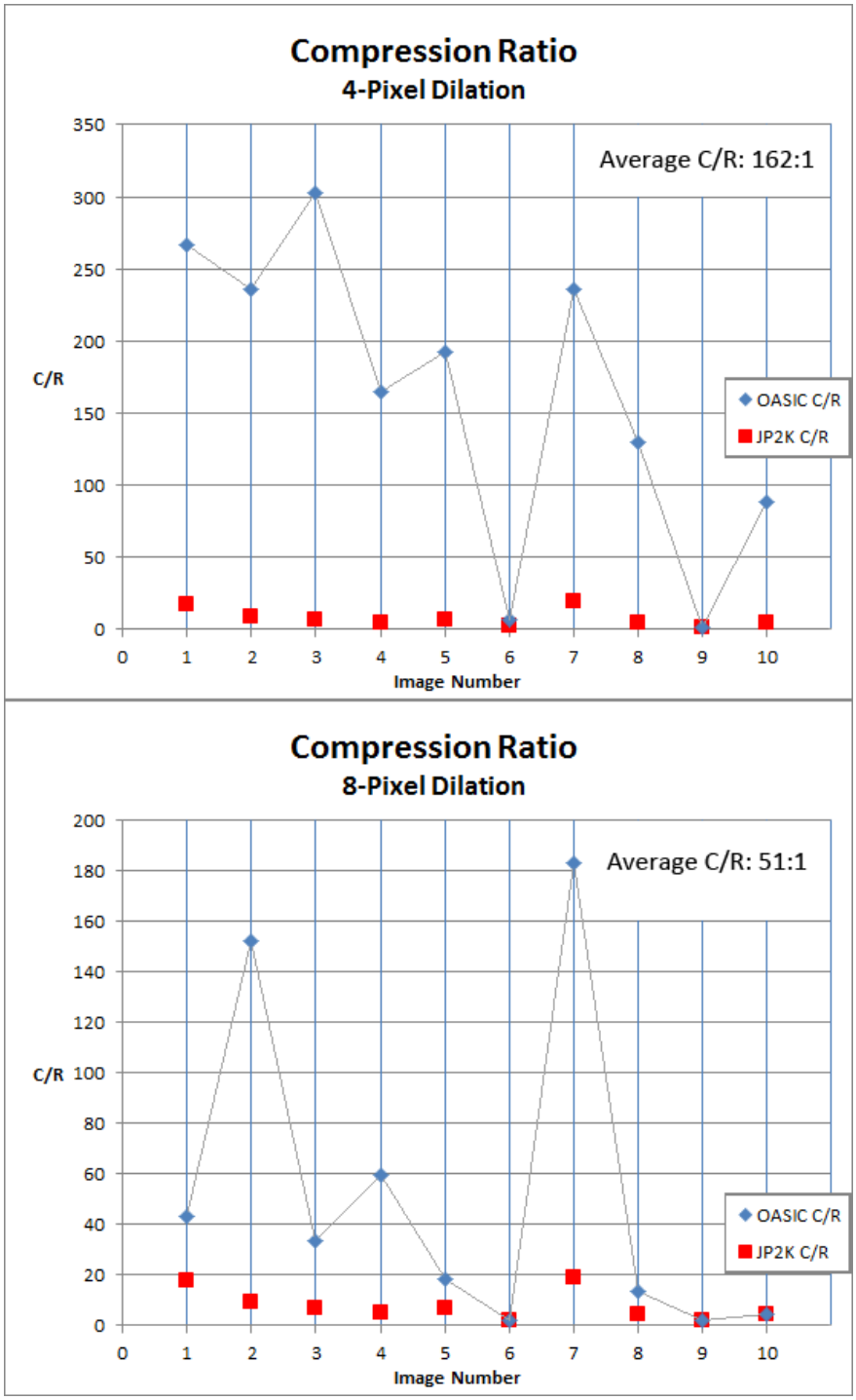

Figure 5.6: Compression Ratio performance of OASIC when compared to lossless JPEG2000 using 10 satellite images with both 4 and 8 pixel dilations.

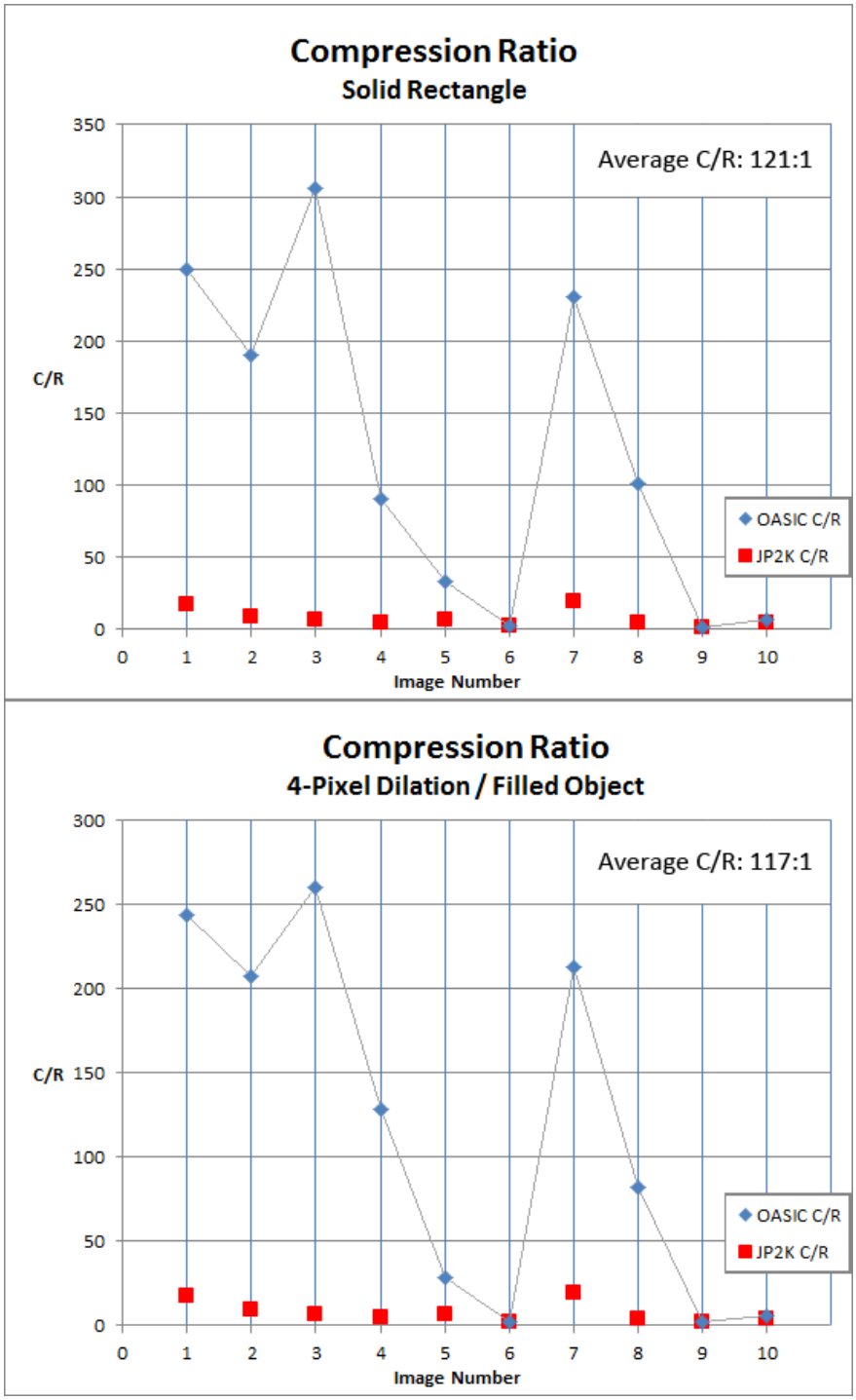

Figure 5.7: Compression Ratio performance of OASIC when compared to lossless JPEG2000 using 10 satellite images with both Solid Rectangle and Filled Object preprocessing methods.

5.3 Fidelity Loss

OASIC's compression algorithm takes advantage of the sparsity of ship pixels in relation to the surrounding ocean. As with all lossy compression algorithms, information must be discarded. For files of comparable size, lossy JPEG2000 and OASIC differ in how they distribute the data loss as demonstrated in Figure 5.8.To illustrate the data loss the original uncompressed image (left) is subtracted from a lossy JPEG2000 image (center) and from an OASIC compressed image (right). The zoomed in areas (inset) indicate the most important difference between the two algorithms: How they distribute noise. The ships detected during OASIC compression experience much less noise than lossy JPEG2000 at the same compression ratio.

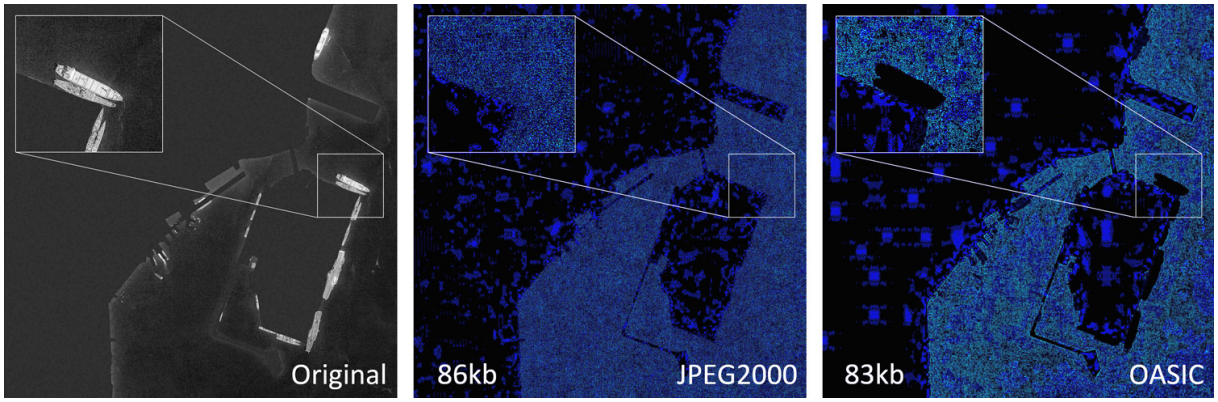

Figure 5.8: The errors are distributed evenly through the ocean and ships with JPEG2000 while with OASIC the ships remain largely error free. Perfectly detected vessels exhibit no error. Errors only occure when ships containing mis-classified (false negative) pixels.

Figure 5.9 and 5.10 display the individual Peak Signal to Noise Ratios for all images. For OASIC, both the overall PSNR and the PSNR for only the ship clusters are shown. A completely noiseless image causes the PSNR to approach infinity, so all graphs are limited to a PSNR of 40dB. In all cases except one (4-pixel dilation, Image 10) OASIC has less noise than any lossless JPEG2000 with the same file size.

Note: Image 10 is nearly entirely obscured by clouds with a single ship. The detection stage classifies over 70% of the image as ship pixels. This phenomena is called over-detection, and is caused by heavy clouds and excessive waves.

A side by side comparison of an OASIC compressed ship, detected at 85%, and a completely undetected ship are shown in Figures 5.11 and 5.12. Note that while both the large and small undetected vessels have lost fine detail, they are still recognizable and are not completely obscured by the lossy background compression.

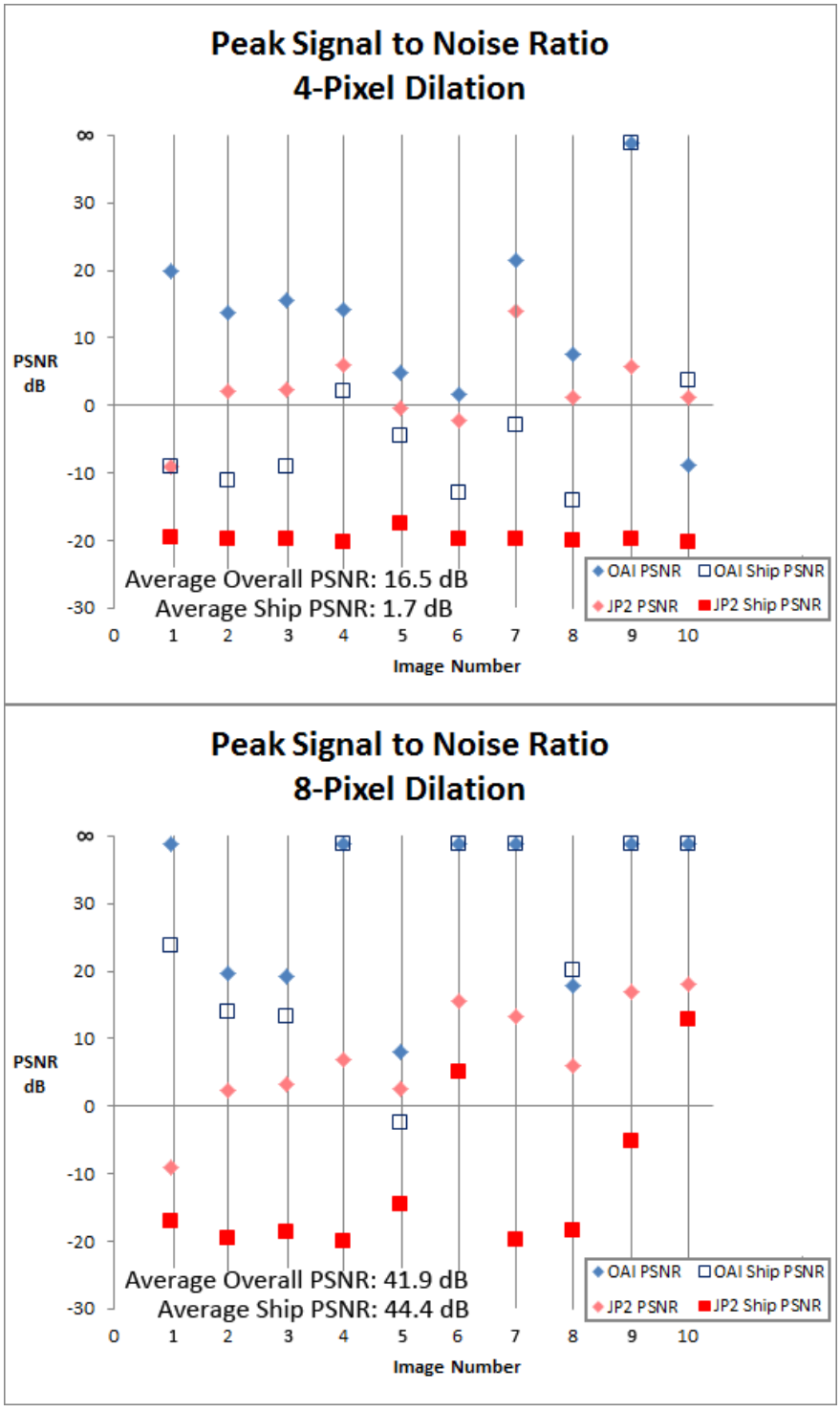

Figure 5.9: These graphs show the relative PSNR levels of OASIC compared to lossy JPEG2000 using 10 satellite images with 4-pixel dilation (Top) and 8-pixel dilation (Bottom)

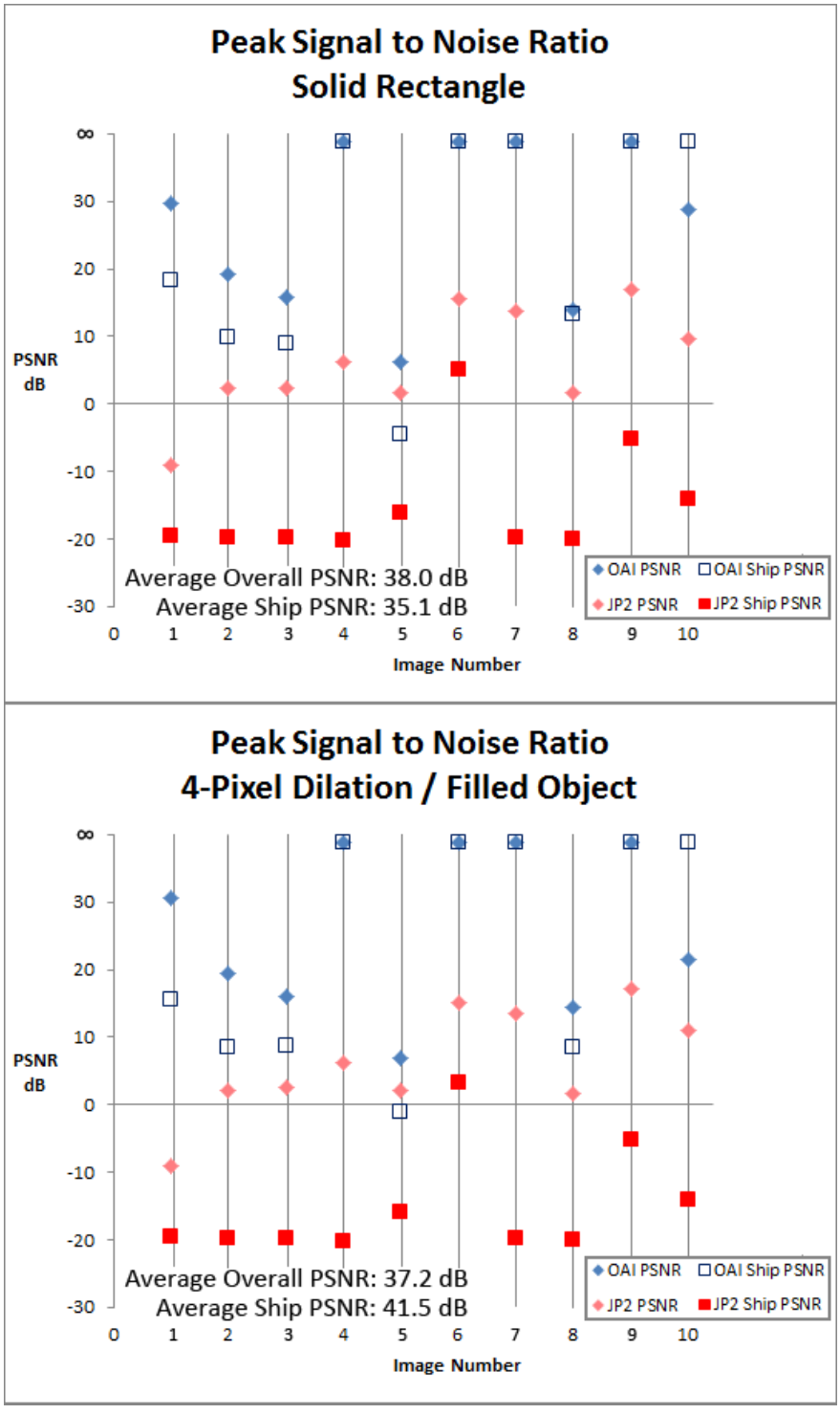

Figure 5.10: These graphs show the relative PSNR levels of OASIC compared to lossy JPEG2000 using 10 satellite images with the Solid Rectangle method (Top) and Filled Object method (Bottom)

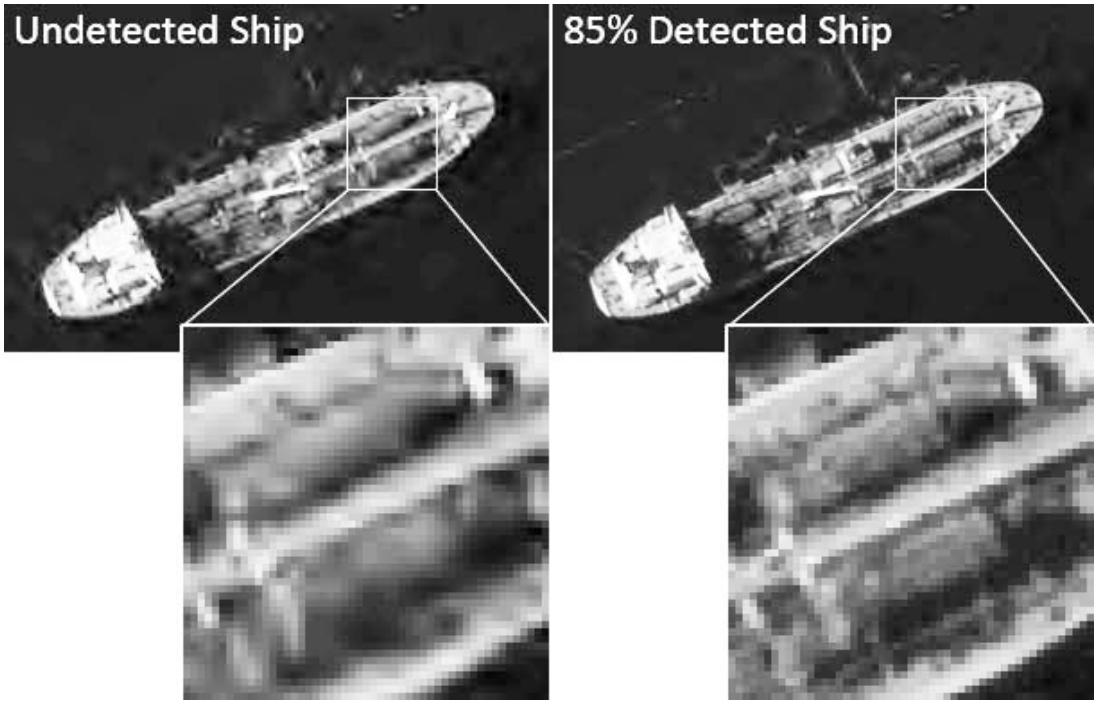

Figure 5.11: Large undetected ships (left) suffer from compression induced noise, and fine details are lost. Even partially detected ships fare better (right).

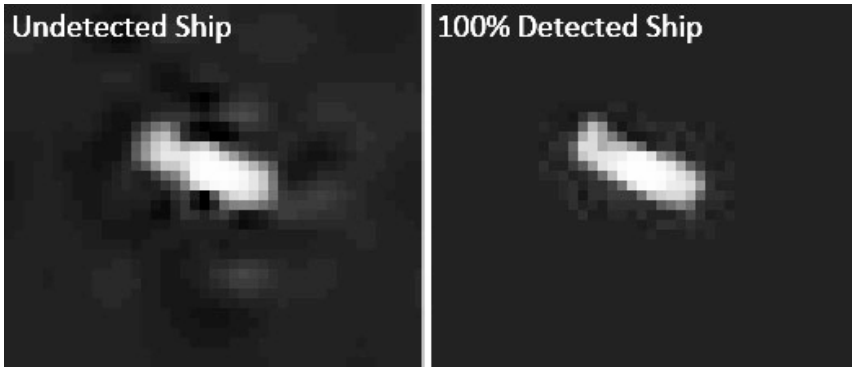

Figure 5.12: Undetected smaller ship (left) and a fully detected ship (right).

5.3.1 Best Configuration

Figure 5.13 displays the relationship between average compression ratios and average PSNR levels. 4-pixel dilation has the most noise due to its relatively low detection rate, despite having an excellent compression ratio. Similarly, 8-pixel dilation has the lowest noise while its compression ratio was the lowest. The Solid Rectangle Method performed the best in terms of total image noise overall. The Filled Object method, however, achieved the second highest PSNR for the ship clusters at an average of 41.5dB and also has a C/R of 117:1 making it the best preprocessing method.

Note: The infinite PSNR values were clipped to 75dB for calculation of the average.

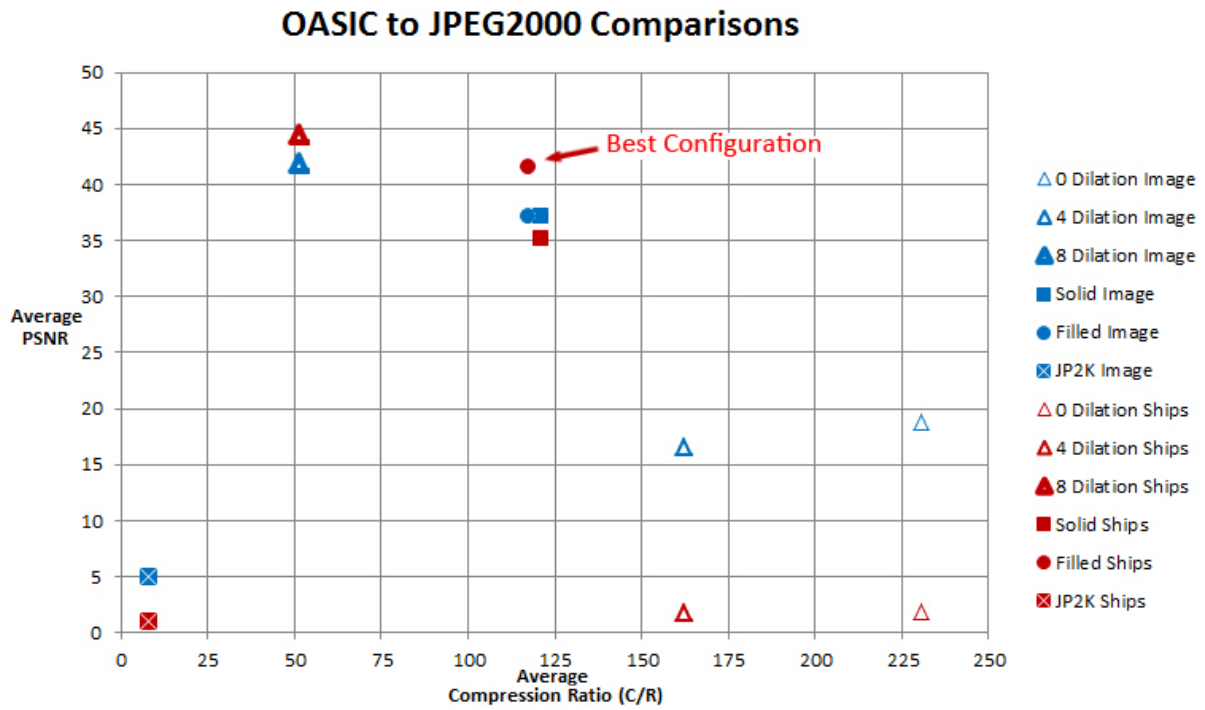

Figure 5.13: The performance of all five preprocessing methods are graphed for both the entire image (blue) and ships only (red). Higher PSNR and higher compression ratios indicate better performance. The best configuration is the Filled Object method.

5.4 Future Research

Research for OASIC has not concluded. A vast range of potential improvements remains, and fertile ground exists for improvement.

5.4.1 Code Optimization

OASIC's implementation in MATLAB does not fully take advantage of capabilities MATLAB provides, many repetitive tasks could be accomplished faster by use of MATLAB's powerful matrix processing operations. In order to eventually use OASIC aboard a satellite as intended, use of other languages should be examined as well as different platforms such as digital signal processors (DSP) and Field-Programmable Gate Arrays (FPGA). The OASIC algorithm takes about 50-90 minutes to compress and store each of the full resolution images. This time varies greatly due to three factors: how many pixels are examined, how many pyramid octaves are used, and how many ships are detected. Preprocessing options have an effect to a lesser extent. Improvements can be made by streamlining the repetitive operations present in both the detection and compression stages.

5.4.2 Automatic Configuration

When OASIC's detector erroneously classifies ocean waves as ships, the number of detections skyrockets. The SVM detection results of each 512×512 tile could be analyzed for this condition and if necessary, the sensitivity reduced, and tile recomputed. Each tile could be analyzed in this way, perhaps adjusting the pyramid configuration as well. Lossy foreground compression could also be evaluated for further compression ratio gains.

5.4.3 Testing and Training Image Set

OASIC only trains on a single image, future research could determine the effects of multiple training images, including rough seas and heavy cloud cover, both environments that caused over-detection. OASIC only tests 8-bit panchromatic images, future research could focus on the use of SAR imagery, multi and hyperspectral images with more than 8 bits per channel. OASIC is limited to 10 high resolution images, and future research could test on many more to better refine performance results.

5.4.4 Other Ship Detectors

OASIC's detection stage is not compared to other ship detectors. Different feature extraction and classification methods may perform better than the DWT and SVM implementation used by OASIC and could permit vast improvements to compression. Future research could focus on comparing current ship detector's to OASIC and what effect adopting better detectors would have on compression performance.

5.4.5 Digital Nautical Charts

The entire image pre-processing step can be automated with the aid of vector-based Digital Nautical Charts. It would require terrain landmarks to be identified and the appropriate DNC to be rectified, (rotated, scaled and adjusted for distortion) before being overlaid over the image.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 6:
Conclusions

6.1 Capabilities

The results of the analysis indicate that OASIC does in fact validate the concept of Content-Aware Adaptive Compression of Satellite Imagery Using Artificial Vision. It outperforms the lossless JPEG2000 format's compression ratio with acceptable loss in fidelity, and it outperforms the lossy JPEG2000's format in fidelity for a file of equal size and compression ratio.

6.1.1 Ship Detection

In 10 images, containing a total of 3014 ship clusters, OASIC's best preprocessing configuration was with using Filled Object, with 4-pixel dilation. This condifuration detected 2947 ships above 50% for a ship detection rate of 84%. Of the nearly 7 million ship pixels in the entire image testing set, OASIC successfully classified 5 million for a total ship pixel detection rate of 72%.

While successful, OASIC also produced a total of 1.4 billion false positive pixels out of approximately 6 billion pixels total. This accounted for 99.8% of the pixels detected. A majority of these false positive pixels are from three large images (6, 9 and 10) that suffered from over-detection, and nearly all pixels in the images were classified as ship pixels. Disregarding the outliers, the false positive rate drops to 76%, over twice as many false positive pixels for every true pixel detected.

Despite the high volume of false positive pixels, the overall compression ratio and PSNR of the images were still very high or at a minimum matching JPEG2000. This is because the false positive pixels tended to be clustered around the ships and not scattered throughout. Many of the false positive pixels near the ships are captured in the same rectangle that would enclose the ship anyway, and therefore incur a minimal loss of compression efficiency, if any.

The efficiency of the Solid Rectangle method is mostly dependent on the orientation of the vessels it encloses and is very inefficient for large ships at diagonal angles. While it guarantees all ship-pixels within its bounds are preserved in the foreground, it does not perform as well as the Filled Object in preserving ships with minimal noise.

The Filled Object method can provide the highest detection rates processing multiple types of vessels of different sizes. This method performs better than Solid Rectangle, and has the most potential for improving the detection rate while having a minimal negative impact on compression ratios.

6.1.2 Image Compression and Fidelity

OASIC when compared to lossless JPEG2000 typically achieved a 17 to 1 compression advantage while achieving an average PSNR above 35dB (nearly flawless.)

OASIC's PSNR fares much better than intuition might dictate, but there is an explanation: Just because a pixel is not detected does not mean it is lost. The lossy background compressor may distort the undetected ship values, but the lower their frequency the less distortion they will sustain. Fortunately, most of the the high frequency pixel clusters (that would suffer the most if not detected) happen to be pixel clusters most likely to be detected.

Suppression of detected objects in the background contributes to OASIC's high PSNR. The lossy JPEG2000 algorithm produces intense ringing artifacts, especially around pixel clusters of high frequency, such as ships. By OASIC suppressing the majority of the ships in the lossy background, these artifacts are generally suppressed as well. Figure 3.14 demonstrates this the best when comparing the pier in (a) and (b) versus (c) and (d).

6.1.3 Summary

In all tests, the worst OASIC performed is equal to lossless JPEG2000. Should the OASIC algorithm be implemented on an imaging satellite, the benefit would be a significant reduction in required channel capacity and time to download an image from space.

Vessels at sea would benefit from this improvement the most: Maritime Domain Awareness, anti-piracy operations, law enforcement at sea and other operations at sea would all benefit from getting the satellite borne intelligence into the hands of the operator faster. Vessels with smaller antennas such as submarines and patrol craft would greatly benefit from OASIC. In the case of submarines, fine-detailed OASIC-compressed satellite imagery of the surrounding ocean could be downloaded quickly, reducing the time the submarine must spend on the surface to access the satellite.

Satellites using OASIC could be engineered to have even larger spatial resolutions and multiple spectral bands with less concern of ever-increasing power and mass requirements.

APPENDIX A:
OAI File

Files compressed by OASIC are stored in an OAI file which begins with a 3 byte header:

[01] Storage Method:
- 'S' - Bounding Rectangle
- 'L' - Two-Layer

[02] Foreground Compression:
- '2' - JPEG2000
- 'J' - JPEG
- 'P' - PNG

[03] Background Compression:
- 'N' - NONE (No Background)
- '2' - JPEG2000
- 'J' - JPEG
- 'P' - PNG

Structure for the Two-Layer method:

[04] FG; 32-bit unsigned Foreground size:
[08+FG] Compressed Foreground image
[09+FG] BG; 32-bit unsigned Background size:
[13+FG] Compressed Background image

Structure for the Bounding Rectangle method:

[04] N; 16-bit unsigned number of sub-images:

In the following six fields x must iterate from 0 to N-1

[06+x*96] xSrc(x); X coordinate in main image
[08+x*96] ySrc(x); Y coordinate in main image
[10+x*96] xSize(x); X size of sub-image
[12+x*96] ySize(x); Y size of sub-image
[14+x*96] xPos(x); X coordinate of sub-image within packed rectangle
[16+x*96] yPos(x); Y coordinate of sub-image within packed rectangle
[16+N*96] PK; 32-bit unsigned Packed Rectangle size:
[20+N*96] Compressed Packed Rectangle
[20+N*96+PK] BG; 32-bit Background size:
[24+N*96+PK] Compressed Background Image

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] F. G. Meyer, A. Z. Averbuch, and J.-O. Stromberg, "Fast adaptive wavelet packet image compression," *Image Processing, IEEE Transactions on*, vol. 9, no. 5, pp. 792–800, 2000.

[2] D. P. Casasent and J. S. Smokelin, "Neural net design of macro gabor wavelet filters for distortion-invariant object detection in clutter," *Optical Engineering*, vol. 33, no. 7, pp. 2264–2271, 1994.

[3] M. Antonini et al., "Image coding using wavelet transform," *Image Processing, IEEE Transactions on*, vol. 1, no. 2, pp. 205–220, 1992.

[4] M. Tello et al., "Use of the multiresolution capability of wavelets for ship detection in sar imagery," in *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*, vol. 6. IEEE, 2004, pp. 4247–4250.

[5] M. U. Selvi and S. S. Kumar, "A novel approach for ship recognition using shape and texture," *International Journal of Advanced Information Technology (IJAIT)*, vol. 2, 2011.

[6] R. N. Strickland and H. I. Hahn, "Wavelet transform methods for object detection and recovery," *Image Processing, IEEE Transactions on*, vol. 6, no. 5, pp. 724–735, 1997.

[7] R. Bogush, S. Maltsev, and A. Aniskovich, "Object detection using wavelet transform." [Online]. Available: www.psu.by/images/stories/fit/personal/bogush/40_prip_2005.pdf

[8] S.-Q. Huang et al., "A novel method for speckle noise reduction and ship target detection in sar images," *Pattern Recognition*, vol. 42, no. 7, pp. 1533–1542, 2009.

[9] A. Kiely et al., "Lossy image compression and stereo ranging quality from mars rovers," *The Interplanetary Network Progress Report*, vol. 42, p. 168, 2007.

[10] C. Zhu et al., "A novel hierarchical method of ship detection from spaceborne optical image based on shape and texture features," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 48, no. 9, pp. 3446–3456, 2010.

[11] W.-C. Fang et al., "A vlsi neural processor for image data compression using self-organization networks," *Neural Networks, IEEE Transactions on*, vol. 3, no. 3, pp. 506–518, 1992.

[12] G. Máttyus, "Near real-time automatic marine vessel detection on optical satellite images," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 1, pp. 233–237, 2013.

[13] K. Rainey and J. Stastny, "Object recognition in ocean imagery using feature selection and compressive sensing," in *Applied Imagery Pattern Recognition Workshop (AIPR), 2011 IEEE*. IEEE, 2011, pp. 1–6.

[14] H. Schneiderman and T. Kanade, "Object detection using the statistics of parts," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 151–177, 2004.

[15] C. Corbane et al., "Fully automated procedure for ship detection using optical satellite imagery," in *Asia-Pacific Remote Sensing*. International Society for Optics and Photonics, 2008, pp. 71 500R–71 500R.

[16] C. Corbane, F. Marre, and M. Petit, "Using spot-5 hrg data in panchromatic mode for operational detection of small ships in tropical area," *Sensors*, vol. 8, no. 5, pp. 2959–2973, 2008.

[17] C. Corbane et al., "A complete processing chain for ship detection using optical satellite imagery," *International Journal of Remote Sensing*, vol. 31, no. 22, pp. 5837–5854, 2010.

[18] M. Degirmenci and S. Ashyralyyev, "Impact crater detection on mars digital elevation and image model," *Middle East Technical University*, 2010.

[19] R. F. Marcia, Z. T. Harmany, and R. M. Willett, "Compressive coded apertures for high-resolution imaging," in *SPIE Photonics Europe*. International Society for Optics and Photonics, 2010, pp. 772 304–772 304.

[20] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The jpeg 2000 still image compression standard," *Signal Processing Magazine, IEEE*, vol. 18, no. 5, pp. 36–58, 2001.

[21] L. He and L. Carin, "Exploiting structure in wavelet-based bayesian compressive sensing," *Signal Processing, IEEE Transactions on*, vol. 57, no. 9, pp. 3488–3497, 2009.

[22] B. Mekisso, A. Talari, and N. Rahnavard, "Unequal compressive imaging," in *MILITARY COMMUNICATIONS CONFERENCE, 2011-MILCOM 2011*. IEEE, 2011, pp. 174–179.

[23] R. E. Korf, "Optimal rectangle packing: New results." in *ICAPS*, 2004, pp. 142–149.

[24] J. E. Fowler, S. Mun, and E. W. Tramel, "Block-based compressed sensing of images and video," *Foundations and Trends® in Signal Processing*, vol. 4, no. 4, pp. 297–416, 2010.

[25] G. Xing et al., "Arbitrarily shaped video object coding by wavelet," in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, vol. 3. IEEE, 2000, pp. 535–538.

[26] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 7, pp. 674–693, 1989.

[27] R. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB*, 2nd ed. Gatesmark, 2009.

[28] T. F. Cootes and C. J. Taylor, “Active shape models—‘smart snakes’,” in *BMVC92*. Springer, 1992, pp. 266–275.

[29] G. K. Wallace, “The jpeg still picture compression standard,” *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[30] D. Duce and T. Boutell, “Portable network graphics (png) specification,” *Information technology ISO/IEC*, vol. 15948, p. 2003, 2003.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia
2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California